Handbook of Multimedia Information Retrieval
Horst Eidenberger

Handbook of Multimedia Information Retrieval

The Common Methods of Audio Retrieval, Biosignal Processing, Content-Based Image Retrieval, Face Recognition, Music Classification, Speech Recognition, Text Retrieval and Video Surveillance

Horst Eidenberger

October 31, 2013

Prof. Dr. Horst Eidenberger
eidenberger@tuwien.ac.at

Interactive Media Systems Group
Vienna University of Technology
1040 Vienna
Austria

International Standard Book Number: 978-3-848-22283-4

Manufactured and published by Books on Demand GmbH, Norderstedt, Germany.

Visit the Web site of this book at *atpress.info/mmir*

Thank you for purchasing this book.

A good laugh
is a mighty good thing,
and rather too scarce a good thing;
the more's the pity.

Herman Melville, Moby Dick, Chapter 5

God keep me from ever completing anything.
This whole book is but a draught –
nay, but the draught of a draught.
Oh, Time, Strength, Cash, and Patience!

ibidem, Chapter 32

Thou canst not tell where one drop of water
or one grain of sand will be to-morrow noon.
And yet with thy impotence thou insultest the sun!
Science! Curse thee, thou vain toy!

ibidem, Chapter 118

Preface

Multimedia information retrieval – or: *media understanding* – is the content-based analysis of audio, bioinformation, biosignals, images, text and video data. In this textbook, we introduce the general concepts of media understanding as well as models and algorithms for feature extraction and categorization. In today's multimedia retrieval science, it is common to work on one type of media only. In consequence, computer vision experts often know little about content-based analysis of audio, biosignal experts are ignorant of text information retrieval, etc. I consider this mutual unawareness disadvantageous to all the named areas of research. There is, for example, a lot that biosignal experts could learn from audio experts and vice versa. Therefore, with this textbook I intend to provide a contribution to bridging the gap of ignorance by comparing methods employed in different domains, emphasizing their communalities and grouping them by fundamental types of strategies.

The entire book has three parts. The first part introduces the big picture of media understanding and covers fundamental concepts of feature extraction, information filtering and categorization. Furthermore, it discusses the practical implementation of media understanding applications and provides a comprehensive list of similarity and distance measures. The second part extends the big picture and covers advanced topics such as local and spectral features, risk minimization, kernel-based methods and dynamic models for categorization. The third part investigates the current frontiers of media understanding science. We review semantic methods for media description, dynamic aspects of filtering and categorization as well as human-like similarity measurement. Eventually, we sketch the neuralization of the bigger picture of media understanding. All three parts together provide a systematic introduction to this field of research by analyzing and clustering of the practically relevant methods.

The approach described in this book originates from my lectures at the Vienna University of Technology where I have taught content-based visual retrieval, audio analysis, text information retrieval and related topics in graduate courses for more than a decade. Over the years, I found the methods employed in the different domains surprisingly similar and started working systematically on iden-

tifying communalities and dissimilarities of the signal processing and machine learning algorithms employed. Feeding the results back into the lectures showed that my students understood the algorithms and applications in the different media domains much better if they had learnt the principal models before.

The book assumes a reader with good knowledge in computer science but not one who has already worked in one of the above-named research domains. Where necessary, references to textbooks are given that cover important foundations of the methods discussed in the book. However, I do not provide a reference where it is sufficient to type the keyword into a web search engine. I recommend you to compare what is written in this book to what other authors write on the web and in related books. Viewing a problem from as many perspectives as possible is always beneficial!

Additional material (exercises, links to software, etc.) can be found at *atpress.info/mmir*. You are invited to make use of the information available on the web page. If you would like to share your experiences and opinions with me, please e-mail to *contact@atpress.info*.

I would like to express my thanks to Christian Breiteneder for supporting me in my work over many years, and to Markus Hörhan, Dalibor Mitrovic, Robert Sorschag, Maia Zaharieva and Matthias Zeppelzauer for many fruitful discussions and for sharing their knowledge with me. Furthermore, I would like to thank my students for their active participation and their interesting contributions. Last not least I would like to thank Ingrid and the kids for more than I can say.

Horst Eidenberger
Vienna, October 31, 2013

Contents

I Fundamental Media Understanding **1**

1 Introduction **3**
1.1 The Problem 3
1.2 Our Approach 7
1.3 Target Audiences 10
1.4 Overview Over the Book 14

2 Applications and Media Types **19**
2.1 Applications of Media Understanding 19
2.2 Properties of Digital Media 24
2.3 Media Description 28
2.4 Media Examples 30

3 The Big Picture of Media Understanding **37**
3.1 Introduction 37
3.2 Elements of Media Understanding 42
3.3 Description of Elements 46
3.4 Application Examples 49

4 Description of Audio and Biosignals **59**
4.1 Introduction and Dimensions of Hearing 59
4.2 Fundamental Audio Transformations 64
4.3 Audio Description by Convolution 70
4.4 Biosignal Feature Transformations 73

5 Description of Visual Media **79**
5.1 Properties of Visual Perception 79
5.2 Color Descriptions 84
5.3 Texture Description 88
5.4 Description by Shapes and Spatial Relationships 92

6 Description of (Quasi-)Symbolic Media **99**
6.1 Symbolic Media Types 99
6.2 Description of Stocks 101
6.3 Description of Text 106
6.4 Description of Bioinformation 112

7 Merging and Filtering of Descriptions **119**
7.1 Merging of Descriptions 119
7.2 Simple Statistical Filtering 125
7.3 Factor Analysis 129
7.4 Understanding Descriptions 133

8 Simple Categorization Methods **139**
8.1 The Setting of Categorization 139
8.2 Rule-Based Categorization 144
8.3 Distance-Based Categorization 146
8.4 Dynamic Association Models 153

9 Probabilistic Categorization **161**
9.1 Foundations of Probability Theory 161
9.2 Independence-based Categorization 166
9.3 Bayesian Networks 170
9.4 Markov Processes 174

10 Application Building **181**
10.1 Application Design 181
10.2 Implementation 188
10.3 Evaluation 193
10.4 Optimization 196

II Professional Media Understanding **199**

11 First Reflection and Bigger Picture **201**
11.1 Conclusions from Fundamental Methods 201
11.2 Building Blocks of Feature Transformations 208
11.3 A Bigger Picture of Media Understanding 213
11.4 Overview Over Advanced Methods 217

12 Transforms in Media Understanding **221**
12.1 Introduction to Unitary Transforms 221
12.2 Transforms with Continuous Bases 225
12.3 Transforms with Limited Bases 230

12.4 Parametric Transforms 237

13 Spectral Descriptions **241**
13.1 Audio Feature Transformations 241
13.2 Biosignal Feature Transformations 248
13.3 Visual Feature Transformations 253
13.4 Spectral Description of Stock Data 257

14 Description of Local Media Properties **261**
14.1 General Localization Methods 261
14.2 Visual Interest Point Detection 266
14.3 Local Descriptions of Visual Media 273
14.4 Local Description of Other Media 280

15 Description of Motion **283**
15.1 Simple Motion Descriptions 283
15.2 Temporal Segmentation 287
15.3 Computation of Optical Flow 290
15.4 Flow-based Motion Descriptions 295

16 Advanced Filtering Models **299**
16.1 Fusion of Descriptions 299
16.2 Selection of Description Elements 303
16.3 Weighting of Description Elements 307
16.4 Advanced Redundancy Elimination 309

17 Principles of Learning Machines **315**
17.1 Introduction to Learning Theory 315
17.2 Concept Theories 322
17.3 Similarity Measures in Categorization 325
17.4 Classifiers in Practice 329

18 Risk Minimization Methods **333**
18.1 Risk Minimization Principles 333
18.2 The Support Vector Machine 337
18.3 Kernel Functions 344
18.4 Advanced Risk Minimization Methods 347

19 Optimization Models **351**
19.1 Fuzzy Similarity Measurement 351
19.2 Learning Meta-Models 353
19.3 Advanced Densities: Mixture Models 359
19.4 From Local to Global Optimization 363

20 Advanced Evaluation **369**
20.1 Cross Validation 369
20.2 Receiver Operating Characteristic Curves 372
20.3 Information-Theoretic Measures 375
20.4 Evaluation of Good Feature Transforms 381

III Frontiers of Media Understanding **387**

21 Reflection of Professional Methods **389**
21.1 Conclusions from Advanced Methods 389
21.2 Building Blocks of Categorization 396
21.3 Which Methods When? 402
21.4 Overview Over Scientific Frontiers 404

22 Media Philosophies **407**
22.1 The Image in Philosophy 407
22.2 Media Theories 410
22.3 Semiotics 415
22.4 Media and Information 419

23 Perception and Psychophysics **425**
23.1 Human Perception and Cognition 425
23.2 Perceptual and Cognitive Errors 431
23.3 Psychophysical Theory 436
23.4 Psychoacoustics and Psychophysics of Vision 439

24 Description by Templates **443**
24.1 Convolution Everywhere 443
24.2 Templates for One-Dimensional Media 447
24.3 Static Visual Templates 452
24.4 Dynamic Template Adaptation Models 456

25 Semantic Descriptions and Applications **459**
25.1 The Semantic Scale 459
25.2 Semantic Feature Transformations 463
25.3 Semantics in Audio, Biosignals and Text 468
25.4 Visual Semantic Applications 472

26 Convergent Filtering **479**
26.1 Models of Convergence 479
26.2 Vector Quantization 484
26.3 The Kalman Filter 487

26.4 Associative Memories 491

27 Frontiers of Learning Machines **497**
27.1 Analysis of Categorization Methods 497
27.2 Limits of Learning 503
27.3 Dynamical Systems Theory 507
27.4 Oscillating Classifiers 514

28 Human-Like Similarity Perception **519**
28.1 Similarity as Measurement 519
28.2 Similarity as Counting 530
28.3 Dual Process Models 536
28.4 Similarity as Alignment and Transformation 541

29 Neural Media Understanding **545**
29.1 Neural Foundations 545
29.2 Artificial Neural Networks 549
29.3 Neural Description and Filtering 554
29.4 Neural Networks for Categorization 557

30 Finale and Future **563**
30.1 Summary 563
30.2 Essential Findings 568
30.3 Critical Review 573
30.4 Outlook: To Do List 576

IV Appendices and Indices **581**

A Mathematical Notation **583**
A.1 Sets and Arrays 583
A.2 Pre-defined Location Sets 583
A.3 Media Templates 584
A.4 Variables 584
A.5 Operations 585
A.6 Building Blocks 586
A.7 Pseudo-Code Format 586
A.8 Some Expressions 586

B Similarity Models **587**
B.1 Quantitative Similarity Measures 587
B.2 Predicate-Based Similarity Measures 588
B.3 Similarity Meta-Models 591

B.4 Dual Process Models 592

C Media Programming Tools **593**
C.1 General Properties 593
C.2 Feature Transformations 594
C.3 Information Filtering and Visualization 594
C.4 Categorization and Evaluation 595
C.5 Mobile Implementation 595

Bibliography **597**

Index **624**

Part I

Fundamental Media Understanding

Chapter 1

Introduction

States the problem of multimedia information retrieval, describes and argues for our approach, introduces the employed notation, discusses the target audiences and gives an overview over the contents, including some suggested paths for particular types of readers.

1.1 The Problem

The three parts of this book provide a thorough introduction into the research areas of computer science that deal with the *content analysis and categorization of digital media*, including audio retrieval, biosignal processing, content-based image retrieval, environmental sound classification, face recognition, genome analysis, music genre classification, speech recognition, technical stock analysis, text retrieval, video analysis and video surveillance, to name a few. We summarize these areas under *multimedia information retrieval* and – more frequently – *media understanding*, since we realize that they share some very important properties:

- They exploit digital signals.
- Signals are summarized by signal processing.
- Summaries are classified by machine learning algorithms.

Digital audio, biosignals, digital images and digital video are data sources that have been investigated in signal processing[1] for many years. Text and bioinformation, on the contrary, are usually not considered appropriate input for signal processing operations. Closer investigation in the following chapters, however, will show that the summarization methods employed on text and, for example, gene strings are comparable to sample-based signal processing operations. In short, multimedia information retrieval aims at the *imitation of the sensual pattern recognition capabilities of the human being.*

Multimedia information retrieval wants to achieve more than just summarization: the computational understanding of media content that is comparable to the understanding of humans. Therefore, machine learning[2] algorithms are employed for the interpretation of digital media summaries. No matter if the data type is audio, image, video, text or some other, machine learning algorithms employ the same learning and classification strategies. Hence, very similar methods are, for example, used in structural alignment of gene sequences and the classification of video events based on prototypes.

Media understanding – our preferred writing below – is not a very popular term in computer science. We choose it because it emphasizes the multimedia aspect of media analysis. The term is derived from the popular *image understanding*, the attempt to analyze images in human-like fashion. Media understanding should do the same to multimedia content (frequently, also *transmedia*). One reason for the limited popularity of the term *media understanding* may be that today hardly any researcher works on the analysis of true multimedia content. Rather, researchers are specialized in image retrieval, computer vision, music retrieval, speech recognition or some other area. Unfortunately, while diving deeply into their focus area, hardly any exchange happens between the research areas and, in consequence, opportunities for mutual stimulation are lost. This exchange is exactly the goal of this publication. See below Section 1.3 for details on this issue.

All multimedia information retrieval disciplines work on digital media, i.e. one- or multi-dimensional data streams of samples perceived through human-like senses (e.g. audio recording, text reading) or more or less sophisticated capturing mechanisms (e.g. ECG electrodes, gene string analysis by gel electrophoresis). Natural or artificial, the media understanding process has to find solutions for a number of fundamental *interpretation and engineering problems.*

[1] For this book, we define signal processing as all reasonable methods of transformation from one signal to another.

[2] In contrast to signal processing, all algorithms that achieve the transformation of limited signals to nameable categories. For the sake of simplicity, we use the terms 'signal processing' and 'machine learning' in a very loosely defined way in the first two chapters of this book. The usage will become more precise in later chapters.

- Polysemy
- Gravity of the sample
- Incomplete categories and magic values
- Curse of dimensionality
- Performance
- Noise, distortions and missing data

Polysemy refers to the fact that, often, media information can be interpreted in more than one way. Simple examples are photos that show more than one motif. For example, a video showing a dog chasing a running person may be interpreted as a police operation or as the recreative activity of a dog owner. The interpretation depends on the *context* of the scene. Polysemy is a particularly hard problem in text retrieval where the meaning of words and sentences depends heavily on the meaning of the text. Dealing with polysemy and context interpretation will be a guiding theme throughout this book.

Gravity of the sample refers to the signal processing aspect of multimedia information retrieval. Many methods employed on audio, image, video and other signals operate on groups of – if not individual – samples, that way losing the context of larger events almost completely. This problem is also referred to as the *semantic gap*. That is the difference in sophistication between the high-level events that human beings easily grasp from media content (e.g. an ECG pulse, faces in an image, the guitar part in a rock song) and the low-level information that computers are able to extract (e.g. the fundamental frequency of a piece of audio). In-between the two levels lie layers of context and interpretation. It is a major endeavor of multimedia information retrieval research to bridge the semantic gap and move from sample-wise signal processing to human-like sophisticated interpretation. We contribute to this end by comparing approaches from separated disciplines and by developing an iterative model of media understanding.

The third problem on the list, *incomplete categories*, is connected to the fact, that most areas of multimedia information retrieval depend heavily on examples provided by humans. In particular, the machine learning component is helpless without good examples for learning. Here, good means well-balanced, comprehensive and differentiated. Non-surprisingly, such examples are hard to provide. The necessary effort is often neglected by computer scientists who are mostly concerned with their models and procedures. However, experience shows that progress in media understanding correlates with good data. One particular danger of incomplete categories is the introduction of *magical values*. If the input

data does not represent the learning problem in the full width and depth, it becomes tempting to use clever quantization, tailor-made transformations, etc. in order to optimize the quality of the results. Needless to say, such solutions generalize very badly. Confronted with an unconsidered case from the same domain the media understanding algorithm fails. It appears that today all too often incomplete categories are exploited by – often, hidden – magical enhancements. Such research, however good the tuned results are, is worthless.

The *curse of dimensionality* problem is not specific for multimedia information retrieval. It stands for the difficulties caused by large sets of parameters that need to be controlled by the investigator. In multimedia information retrieval, operational blocks of signal processing and machine learning are combined to solve particular media recognition problems. If each method requires a few parameters we soon end up with hardly handleable sets of degrees of freedom. In consequence, testing and evaluation in media understanding consume a significant share of the time, because every new parameter multiplies the dimensionality of the problem. Simplification by the elimination of parameters is a must in media retrieval.

So is *performance optimization* since large amounts of data need to be processed by complex operations. For example, the indexing of all goals scored in one season of the English Premier League amounts to the investigation of 34200 minutes of high-definition video. Still, this is a small application compared to automatic video surveillance of a large city such as Paris. Similarly, structural alignment of gene sequences requires considerable resources. Since the processing power of even the largest supercomputer is limited (and very expensive) algorithmic optimization is the only way to do sophisticated multimedia information retrieval.

Eventually, dealing with *noise, distortions and missing data* is a practical problem of media understanding. Noise is almost ever present in digital media simply because the sensing process is particularly prone to noise. Occlusions are obvious in visual material (e.g. identifying faces when sometimes features are occluded by beards or sunglasses) but do, likewise, occur in other media types. For example, certain sounds are masked by others and lost. Some words are are ignored during reading, etc.[3] Like the other fundamental problems the handling of noise and distortions is a recurring topic of media understanding.

In summary, media understanding spans an umbrella over a large group of research disciplines that deal with the summarization and interpretation of digital media content. The objective is to imitate humans as good as possible with the benefits that computers are cheap and do not get tired. Unfortunately, the state-of-the-art in multimedia information retrieval is – in most areas – still far off the goal. Current media understanding applications can only assist human

[3]Did you spot the second *are* in the last sentence?

classification of digital media. The aim of this book is to bridge the gaps between the fields of media understanding and to enable learning from each other's best practices.

In the remainder of this chapter, we discuss our approach to the formulation of a unified theory of media understanding (next section). Section 1.3 reflects the intended audiences of the book and sketches some paths through the book for important groups of readers. The last section gives an overview over all parts and chapters of the book with the intention to show that we move in a step-by-step manner from simple to sophisticated multimedia information retrieval problems.

1.2 Our Approach

In this section, we discuss our approach to integrate the various areas of research summarized as media understanding. We discuss the goals connected to the integration process, the obstacles on the way and the positive and negative results caused by it. In the last part of the section, we reflect the form of argumentation: Should it be mathematical or statistical?

Foremost intention of the three parts of this book is to deliver an overview over the majority of methods employed in the media understanding areas of research. Besides explaining the intentions connected to each method and the process in which it is embedded, our focus is on stressing the *communalities and differences of the methods*. Critical reflection of the similarities of methods employed in two or more fields – but likewise, within a field – should have a positive influence on the learning process. Eventually, the reader should become able to understand the degrees of freedom that exist in multimedia information retrieval – independent if the domain is video, audio or some other data type. By understanding the analysis of all major data types, the reader is enabled to implement media understanding applications on multimedia content as well as on single-media content.

It will turn out that the same basic operations are employed in the procedures designed for music genre classification as for biosignal detection. The paramount principle of this textbook can be formulated as applying signal processing on the signal processing methods used in multimedia retrieval in order to summarize them and to show that, actually, these summaries are very similar for the fields of research investigated. Furthermore, we do a classification of the classification methods of media understanding in order to show that not the data type is decisive for the selection of a particular machine learning method but the availability of human judgment on the data. That is, we apply media understanding on itself for the benefit of understanding how it is being done, which methods reappear frequently and which strategies are employed for fine-tuning.

Brainstorming the last paragraph – so to say, the vision statement of this textbook – we come up with two major difficulties:

- Understandability versus preciseness
- One mathematical language versus many

Our intention is to provide the reader with a clear understanding of the concepts of media understanding. However, media understanding is as an independent method, as mentioned in the previous section, hardly existent today. Rather, researchers focus on one particular data type and try to optimize their results by fine-tuning their procedures. It is a widespread belief that genuine methods are employed in the individual fields. For example, audio specialists pay attention to developing their methods further but do usually not pay attention to what is going on in the video domain, and vice versa. We consider this mutual ignorance unfortunate since many potentials lie in the communalities of these – only at first sight fundamentally different – approaches.

However, it is impossible to achieve the desired understanding and to build bridges between the areas of research while delivering the same degree of preciseness as a textbook for one particular problem would. We have to give up a bit of preciseness for the benefit of uncovering similarities between the methods. Below, we will argue for our approach with the benefits of general understanding. For the moment, let us state that we aim at the development of a *harmonic, i.e. conflict-free*, scientific system over all disciplines of multimedia information retrieval. We accept a certain loss of preciseness in the explanation of the individual methods and procedures for the benefit of a general theory.[4]

Doubting that different data types would require fundamentally different approaches, we believe that the status quo of multimedia information retrieval has been reached by a tendency of researchers for ignoring what is going on outside their immediate environment. If this hypothesis is correct, we may uncover in due process principles and potentials true for more than one area of media understanding. Principles are models, methods, algorithms that work in the same manner (except some fine-tuning and magic) in more than one discipline while potentials are principles successfully applied in one area but so far ignored in others. We are positive that these benefits exceed the loss from a less precise description of the individual methods.

However, one problem is hidden in our determination to develop a conflict-free system. It becomes visible when we approach the problem from the game-theoretic point of view. Describing the slightly different components of related systems is like having different interpretations competing for the rank of the best

[4] Alfred North Whitehead formulated this principle in his *Introduction to Mathematics* as: *"To see what is general in what is particular and what is permanent in what is transitory is the aim of scientific thought."*

explanation. In such a game usually more than one Nash equilibrium exists. Unfortunately, Nash equilibria, though conflict-free, may take awkward forms. It is, therefore, not enough for us to search for some harmonic description of the methods employed in more than one area of multimedia information retrieval research. Moreover, we have to mind that the chosen explanation represents all instances of usage in a reasonable way.

The second major difficulty of our approach is defining a common mathematical language of media understanding. Biosignal processing, audio retrieval, computer vision and the other fields under consideration have developed sophisticated notations that are to a large degree incomprehensible to the non-expert. On closer examination, it appears, however, that the differences are not fundamental but mostly founded in the employed notation. It is, therefore, tempting to try a unification of the individual notation systems. Mutual understanding requires recognizing one's own concepts in the other field. Concept recognition requires understanding the language used for description.

Hence, we introduce a unified notation for media understanding. The full details can be found in Appendix A. The major concepts are introduced in the next two chapters (media representation, media processing steps). Further elements are introduced on first occurrence. The general idea of our notation is that algorithmic concepts are preferred from mathematic concepts. The reasons for this decision are given in the last part of this section. Practically, media objects are represented by arrays, many functions hold the upper hand over few operators and employed names are pre-defined for the usage of variables (e.g. weights) and constants.

One principal advantage of the common notation is that once learnt it can be applied to all other methods and areas of research in the same fashion. Certain data structures and transformations are clear on first sight. Misinterpretations are avoided, and a steep learning curve is achieved. The disadvantage to this advantage is that a new notation has to be learnt. The notations used, for example, in image understanding are very sophisticated. There is no need for the image specialist to learn a new notation. The available one can be used for everything that is required. However, we target at the image specialist who is interested in what is going on in the biosignal or audio domain. There, different notations are employed that would hinder the quick apprehension. It is an advantage of the common notation that it eliminates this problem. Furthermore, the notation may contribute to the overall goal of identifying potentials in some areas of research that are not fully exploited yet. If certain functions, variables, etc. are not used in one area it should be questioned, why? For some technical reason or just out of ignorance? The common notation enables answering such questions rapidly.

Eventually, the understandability versus preciseness problem should also be discussed from the point of view of the common notation. Certainly, the nota-

tion is – for good reasons – not sufficiently differentiated for allowing to express all details of a particular method. On the contrary, certain aspects of fine-tuning cannot be expressed in this mathematical system. For this loss of preciseness, we gain the guarantee that similarities hidden in methods can more easily be identified. Practical experience supports this view. It is a frequent experience in multimedia information retrieval that the principal model accounts for approximately 80 per cent of the quality of some method while fine-tuning accounts for 20 per cent or less. This 80:20 rule appears in various forms in multimedia retrieval, e.g. signal to noise relationships in biosignals, fundamental frequency of music versus overtone structures, fundamental face features to high-frequency information, object shapes to textures, etc. With the common notation we focus on the 80 per cent communalities while giving up the 20 per cent differences for the benefit of better understanding.

Apart from vision and notation, one further issue requires discussion in this section. It is the question how the presented processes and methods are argued for. Of course, all the demonstrated algorithms are practically used in one or more media understanding disciplines. Methods (e.g. from signal processing) ignored in all application areas are – mostly – not discussed below. However, practical usage is not a sufficient reason for inclusion in a scientific textbook. Two lines of argumentation are thinkable:

- Mathematical correctness
- Practical successfulness

Following the mathematical correctness argumentation methods have to be justified by proofs. Attempts to prove signal processing methods and machine learning methods can be encountered frequently in the literature. However, the mathematical proof is not a sufficient justification for the application of some method in media understanding. What counts is the *ability of a method to imitate human classification behavior as good as possible*. The practical successfulness of a method is detected by statistical analysis and evaluation against human behavior. Multimedia information retrieval is experimental engineering. In consequence, the reader will find no proofs in this book. Instead, we describe the conditions under which a particular method has proven successful experimentally. In summary, our argumentation is – where such a choice can be made – always statistical instead of mathematical. This point of view is in line with the algorithmic design of the used notation.

1.3 Target Audiences

The primary audiences of this textbook are graduate students in computer science with an interest in digital media analysis (see below for a more differenti-

ated list). The book was developed as a course book for a twelve-hour Bologna course module with lectures and exercises on media understanding. In the computer science curriculum, this module should follow introductory and advanced courses on the individual areas of research, most importantly, pattern recognition, computer vision, image understanding, video analysis, audio analysis and text information retrieval. The module provides additional material on these subjects as well as on other subjects like medical informatics (bioinformation processing, biosignal processing), computational intelligence (advanced and experimental machine learning methods) and visual computing (multimedia applications of media understanding such as augmented reality). Most importantly, it clarifies communalities and differences of the methods employed in the individual fields thus providing a workbench of tools for media understanding in less well-known areas. In the more than ten years of teaching at his university the author has frequently experienced that graduate students – after visiting the obligatory courses – are very well able to apply media understanding methods on audiovisual content with good results. They are, however, to a much lesser degree capable to reflect the employed methods and therefore, almost unable to develop the method set further. The media understanding module takes this deficit on by the approach outlined above.

Requirements to the reader include undergraduate knowledge of linear algebra, analysis, statistics (including optimization) and general computer engineering (programming, data structures, etc.). Experience with the processing of digital media is of benefit though not mandatory. Neither is prior knowledge of one or more fields of multimedia information retrieval required.

In detail, this book is intended for the following groups of readers:

- Graduate students in computer science (visual computing, medical informatics, computational intelligence, etc.)
- Audio experts (speech recognition, music classification, etc.)
- Bioinformation experts (e.g. gene alignment)
- Biosignal experts (EEG analysis, ECG analysis, etc.)
- Finance data analysts (e.g. stock analysis)
- Information retrieval experts (e.g. text recognition)
- Vision experts (image retrieval, video event classification, etc.)

Graduate students fall in one of two groups: those with prior knowledge in one or more fields of media understanding and those without. For the first group, the main advantage of the book lies in the introduction to the method set

applied in related areas. Existing models are reviewed critically and potentials can be uncovered. Likewise, new fields of application for existing knowledge are opened at minimal effort. For the beginner, the book provides an overview over the various aspects of multimedia information retrieval, an exhaustive list of employed methods and a unified view on open problems that allow for critical judgment of the value of this discipline. Students from a signal-oriented subject (e.g. visual computing) hear about the learning-oriented side of the problem. Students from learning-oriented subjects (e.g. computational intelligence) come to know the details of applications and of signal-oriented problems. For both groups seeing the other side should enable them to develop a fuller understanding of their own domain. For example, why do populations of descriptions look the way they look? Why do particular learning algorithms fail on particular types of descriptions? Etc. This audience should benefit from the entire book.

Audio retrieval experts are another key audience of the book. Working with audio requires excellent knowledge in signal processing and, at least since recently, improved knowledge in machine learning. However, some methods are very popular while others are literally unknown. Frequently, these other methods are accepted in related areas such as biosignal processing or visual analysis. We suspect that the tradition of audio analysis (in particular, in such well-investigated fields as speech recognition) has a stronger influence on method choice than objective comparison of and selection from the available range of methods. The book intends to show audio experts structural similarities to their methodology in related fields. Many intriguing similarities do exist. The audio expert does not have a far way to go in order to become a media understanding expert.

The imagined bioinformation expert has excellent knowledge of similarity measurement techniques, structural alignment and process optimization. Signal processing plays little to no role in bioinformation processing. We see four major benefits from reading this book for the bioinformation expert. Firstly, other areas (in particular, text retrieval) know many related similarity measurement methods that may also be applicable on bioinformation. Secondly, results from psychological similarity research that are today already applied in the visual analysis could as well be beneficial to the domain. Thirdly, machine learning provides many more models than the Bayesian procedures usually employed for sophisticated gene string matching. Such methods are, for example, used in visual object recognition. Eventually, considering the length and complexity of biodata it may pay off to summarize them – in a similar fashion as text – in order to make them processable more easily. For all known species, the majority of gene data are junk. Cleverly adapted signal processing could help to speed up the analysis process.

Biosignal experts have a similar background as audio specialists – in the signal processing domain. However, classification by machine learning plays a

surprisingly small part in biosignal processing today. One frequently mentioned argument is that computational classification is too error-prone and risky. Therefore, the classification step is left to the user. This argumentation is, without doubt, reasonable. Still, machine learning may play an important role in the *preprocessing* of biosignal events (e.g. automated warnings of abnormal ECG data). This book provides the biosignal expert with the knowledge required to make the step from signal processing to automated classification as it is performed on audio and visual media today.

Finance data analysts usually have at least a limited understanding of signal processing though surprisingly few methods are automated in this discipline. Data summaries such as peaks, holds, etc. are mostly generated by hand and events are classified by looking at them. Clearly, such a proceeding opens the door for overbearing subjectivity. Humans are highly gifted in arguing for some theory while the objective value of it may be small. Automated procedures for chart analysis that are based on pre-defined building blocks that are also used in related areas should help to elevate the level of quality as well as the level of seriousness in this discipline. Stock analysis experts should, in particular, benefit from visual template matching models and from probabilistic classification methods.

Information retrieval experts are mostly concerned with the processing and understanding of text. Like in bioinformation analysis researchers focus on the machine learning aspect. Sophisticated methods were developed for analysis on the word, sentence and text level. On the other hand, signal processing aspects are of little to no relevance in information retrieval today. Automatic summarization is used, but mostly based on classification. We believe that text experts could benefit from the ideas developed in audio and video summarization as well as from the similarity models, for example, employed in video event classification. Audio and video analysis originate partially in text information retrieval but have matured to a degree where some of the results of both disciplines may be useful inspiration to the original problem domain.

Eventually, vision experts (specialized in image analysis or video analysis) are comparable to audio experts in their thorough knowledge of the signal processing side as well as the machine learning side of multimedia information retrieval. However, they usually know surprisingly little about what is going on outside their domain (image/video, but certainly not audio) even though insights from these areas could be of the highest relevance for their field. The major benefit of this book to the visual expert is to see the other side of the wall, in particular, how the same methods of signal processing are employed on audio and biosignal data and how the same machine learning methods are employed on genes, text and audio.

In conclusion, we provide a broad review of the methods employed in multimedia information retrieval – hopefully – for the benefit of all groups of experts

in the various domains of application. More details on the media under considerations and their needs in terms of signal processing and machine learning are given in the next chapter. The message of this section is that by reflecting the state-of-the-art we endeavor to uncover potentials for improvement in media understanding.

1.4 Overview Over the Book

We would like to close the introduction with an outline of the *chapter structure of all three parts of the book* and recommendations of paths through it for the groups of readers sketched above. Not all chapters are interesting for all target audiences. However, we hope that the book contains substantial information for each reader.

This textbook is structured in three parts. Each part covers a two-hour lecture on media understanding. The second part is based on the first part. The third part is based on the second part. For beginners, it is recommendable to work through the first part before moving to the advanced chapters. Domain experts may go directly to the more sophisticated topics. Each part is organized according to the typical flow of information in media understanding applications. In Chapter 3 we introduce this sequence of signal processing operations and machine learning operations. Hence, the first chapters of each part are on media properties and signal processing of digital media while the later chapters are on machine learning of digital media descriptions.

The big picture of media understanding is only represented on the part level. Chapters contain information on related methods, either on signal processing or machine learning. Related methods are referred to as building blocks (e.g. integral transformations, statistical filtering methods, probabilistic learning). Building blocks emerge from the reflection process discussed above. The common notation supports the description process of building blocks. Where necessary, cross-references between related building blocks are made. In particular, Chapters 3, 11 and 21 reflect the overall structure of the media understanding process and paint the big picture of data manipulation and information flow.

Of the remaining chapters of the first part, Chapter 2 discusses the properties of the media types under consideration. Along the reflection of communalities and differences of the media types, the notation for media objects is introduced. Furthermore, visual examples of media objects are given and discussed.

Chapters 4 to 6 deal with fundamental signal processing operations employed for the summarization of single-media data. The two initial chapters introduce the most common and easy to comprehend methods used today on audio and video. Audio analysis is discussed first, because the data type is the least complex of those under consideration and a large number of signal processing meth-

ods have been proposed. In the following visual chapter we will see that some methods employed on visual data follow the same ideas as those employed on audio. In the last chapter of the group signal processing operations employed on non-audiovisual data are reviewed. We will see that only few genuine methods emerge. Most ideas are shared with signal processing of audiovisual data.

On the passage from signal processing topics to machine learning topics Chapter 7 introduces important concepts for the statistical transformation of extracted media summaries. These methods follow two purposes: simplification of the summaries and elimination of noise. We introduce the basic filtering concepts in this chapter. Advanced methods are discussed in the second part in Chapter 16.

The following Chapters 8 and 9 introduce the fundamental concepts of categorization by machine learning. Theoretical models are reviewed, and simple concepts are evaluated. In the second chapter of this group, we discuss probabilistic methods for categorization. Not all of these models are easy to comprehend. However, they constitute the foundation for a number of advanced methods and must, therefore, be presented early. Furthermore, probabilistic categorization methods can be implemented with limited effort which makes them suitable for simple though well-performing media understanding applications.

The final chapter of the first part deals with the implementation of applications. Media understanding is an experimental discipline. All methods presented in this book are suitable for application. This chapter explains how media understanding applications are built and tested.

The second part follows the structure of media understanding introduced in the first part. Many methods introduced in this part are based on the concepts introduced in the first part. Like the first part, the second comprises a two hour media understanding lecture – here on advanced concepts.

In Chapter 11, the insights gained from the first part are critically reviewed, and the big picture is revised and extended. This bigger picture comprises a more detailed view on the machine learning process and a number of feedback loops between the steps of the process. The bigger picture of media understanding provides the foundation of the second part of the book.

Chapters 12 to 15 deal with signal processing issues. This time, the chapters are not divided by media type but by purpose. Chapter 12 reviews transforms that are, likewise, employed on audio, visual and biosignal content. The underlying convolution operations have already been introduced in the first part. This chapter extends and reflects their understanding. The next chapter of the group introduces signal summarization techniques that are based on integral transforms. Such methods are, likewise, used in the audiovisual domain, biosignal processing and even technical stock analysis. Chapter 14 introduces summarization methods that are employed on all data types. These methods appear at first very heterogeneous but closer inspection shows that they share some prop-

erties. The last chapter of the group deals with an aspect only present in video: motion. This property emerges from the dimensionality of the underlying media type. Since the resulting signal processing operations are partially unique we put them all in one chapter.

Similarly to the first part, we proceed from signal processing to machine learning by a chapter on information filtering. Chapter 16 introduces advanced concepts of redundancy elimination and dynamic filtering processes.

The Chapters 17 to 20 deal with various advanced aspects and models of machine learning. The first chapter of the group reviews the content of Chapter 8 and introduces important concepts from human learning psychology. On this foundation, the fundamental machine learning concepts are generalized. All concepts and models required in the subsequent chapters are introduced in this place. The two following chapters deal with practically usable information. Sophisticated and powerful machine learning methods are introduced and their application on media summaries is explained. Chapter 18 deals with categorization by risk minimization. Chapter 19 introduces meta-processes for categorization. We will encounter methods for clustering by separation as well as clustering by hedging. In the process of discussion, we will review the potentials and limitations of the methods. Eventually, Chapter 20 is dedicated to the evaluation of the efficiency of the methods introduced in the prior chapters. This chapter is based on the last chapter of the first part, but extends the set of methods by some fundamental tools that are highly useful for practical application.

The third part is organized along the big picture of media understanding like the first two parts, and it is designed to be one sophisticated lecture on top of the advanced concepts. Still, one major difference to the two other parts exists: Some of the chapters do not introduce methods intended for direct practical application. Rather, these chapters reflect the information of the earlier chapters and extend it towards scientific frontiers of media understanding. This part is intended for the media understanding researcher looking for inspiration for future research. Nevertheless, some chapters also describe practical tools for the enhancement of media understanding applications as well as techniques alternative to those introduced in earlier chapters.

The leading Chapter 21 reflects the results gained from the first two parts of the book. Building blocks of feature transformations are generalized and extended, new building blocks of categorization methods are identified. The next two Chapters 22 and 23 discuss media-related issues. The first chapter introduces results from semiotics, media theory and related areas of research that are potentially helpful in the media understanding process. This chapter processes information that is soft in comparison to most other chapters. Still, we believe that the philosophical domain provides a number of fruitful stimuli for media understanding. In the second chapter, psychophysics and perception are discussed and their influence on signal processing is outlined. Partially,

psychoacoustic knowledge has already been employed in earlier chapters. This chapter summarizes the results of this research discipline and discusses influences on media understanding.

Chapters 24 and 25 attribute signal processing problems. Chapter 24 deals with template matching, a group of operations equally important in all considered domains. Valuable methods are explained and abstracted into a general model of template matching by crosscorrelation. The second chapter reflects so-called semantic descriptions, i.e. signal summaries that are derived from other summaries. The intention is to imitate the human cognition process that derives advanced concepts from simpler ones and from sensual information. We discuss the potentials and limits of semantic signal processing operations.

Chapter 26 is another information filtering chapter. This time, mostly theoretically relevant models are discussed and unified into a general model. One exception of the highest practical value is the Kalman filter that is introduced and critically reviewed.

The machine learning group of the second part comprises Chapters 27 to 29. The first of these three chapters deals with theoretical models for the identification of the boundaries of machine learning. Existing theories are compared and general conclusions are drawn. The second chapter provides an overview over the results of psychological research on human similarity perception. The practical results of this research are summarized in Appendix B. We believe that the insights gained from psychological research are highly beneficial for sophisticated machine learning in media understanding. After all, media understanding endeavors to imitate human behavior. The last chapter of the group follows this line of argumentation and discusses advanced neural models for machine learning. Some of these models have practical relevance, since they can effectively be implemented in computer systems.

Chapter 30 summarizes the most important results of all three parts and makes them subject to a final reflection process. From the state-of-the-art conclusions on the near future of media understanding are drawn.

Before we conclude this chapter, we would like to sketch some paths through this textbook for the before-mentioned audiences. Obviously, the first and second part are of primary interest for the beginner. Chapters 3, 11 and 21 should be beneficial for all target audiences. Likewise, should the three information filtering chapters be of interest to all groups of readers. Furthermore, we recommend reading the following chapters to the specific groups of readers:

- *Audio expert*: Chapters 6, 14 and 24 should provide interesting additional knowledge on summarization methods. Chapters 22, 23 should be of general interest to audio experts. Chapters 27, 28 and 29 should contain some interesting novelties from the machine learning domain.
- *Bioinformation expert*: Experts from this audience usually have good

knowledge in probabilistic machine learning. However, other algorithms are often less well-known. We recommend Chapters 18, 19 and 29 from the machine learning domain. Some aspects of signal processing may also be of interest. We recommend Chapters 6, 14 and 24. Furthermore, Chapters 22, 23, 19 and 28 should be of interest to this audience.

- *Biosignal expert*: Recommended are all chapters on machine learning from all three parts, since this part of media understanding is often neglected in biosignal processing. Furthermore, Chapters 22, 23 should be of general interest. Chapters 14, 24 and 25 should provide interesting information on advanced signal processing.

- *Finance data analyst*: Almost all chapters should be of interest to the finance data analyst. In particular, we recommend the first part on fundamental media understanding. From the two latter parts Chapters 14, 18, 24 and 28 should be of special interest.

- *Information retrieval expert*: We consider the text retrieval expert firm in all aspects of machine learning. However, some aspects of signal processing may be of interest. We recommend Chapters 6, 14 and 24. Furthermore, Chapters 22, 23, 19 and 28 should be of interest to this audience.

- *Vision expert*: This expert is in a similar situation as the audio expert. We recommend Chapters 4, 24 and 25 from the signal processing domain. Chapters 22, 23 should be of general interest. Chapters 19, 28 and 29 may provide information not so well-known in this domain.

In summary, the three parts of this textbook guide the beginner from zero to expert knowledge in the various domains of media understanding. Concepts and procedures are introduced, reviewed, abstracted and generalized into building blocks. Eventually, a workbench of methods is provided for the analysis and design of media understanding systems. Exercises and additional material can be found on the web.[5]

[5]Please visit *atpress.info*.

Chapter 2

Applications and Media Types

Lists and discusses important applications of media understanding, characterizes the fundamental media types, names and lists their major properties, compares media types by these properties, provides a formal notation of media objects and introduces the leading example.

2.1 Applications of Media Understanding

This chapter is dedicated to the media that constitute the foundation of all media understanding applications. We start with an overview over important applications, briefly discuss their components and arrive at the fundamental media types of media understanding. In the second section, the essential properties of the media types are investigated. Media dimensions and bandwidth requirements[1] are discussed. Then, the representation of media in the mathematical system along with important operators is introduced. This introduction is continued in the next chapter. The last section introduces examples that lead the methodological discussion throughout the book. Eventually, a simple media understanding process is used to compare the most general properties of the media types.

Table 2.1 lists some important applications of media understanding along with some characteristically employed media types, signal processing operations

[1] Here, generic for the size of a media event.

and machine learning operations. The list should illustrate how diverse media understanding is. It is astonishing that most of these applications share the same methods. The list is not exhaustive nor are the listed methods. In fact, most methods mentioned in the table have been employed on all the listed application problems. Still, the selected methods are among the best-performing ones. It is noteworthy that only digital media are given in the media column. This selection follows the definition of media understanding in the previous chapter. We do not consider this choice a limitation since today hardly any media signals are captured, processed, broadcasted or stored analog any more. The quality of digital sampling and redundancy elimination has arrived at a level where any signal considered in media understanding can be represented digitally at satisfactory quality.

Most applications should be self-explanatory. Some special applications are concept recognition, copy detection, flow detection, P300 detection, query by humming and unusual event detection. Concept recognition aims at the association of (mostly, visual) media with names, e.g. the association of animal photographs with the name 'cat.' Copy detection is important in image analysis. Such applications try to identify the, for example, copyrighted original of a given media sample. Flow detection is a classic application in the social sciences and emerging in computer science. Here, the goal is to determine typical path structures of moving humans. Flow analysis may be embedded in path planning or the psychological analysis of human behavior. P300 detection comes from biosignal detection. It aims at the identification of peaks of brain activity that usually emerge 300ms after an unusual stimulus has been presented in a sequence of well-known stimuli. Query by humming is motivated by the desire to retrieve the name of a piece of music by simply humming it. This problem has turned out to be among the hardest audio understanding problems of the last decade. Eventually, unusual event detection is, for example, employed in video surveillance for the detection of emerging situations.

All applications operate on one or more of the following data types: audio, image, video (sensual media), biosignals, stocks (artificial time series), text and bioinformation. The latter two data types are, as we will see below, fundamentally different from the others which makes their signal processing a different task. Machine learning, however, is hardly different for text and bioinformation from the other data types.

Before we continue with the analysis and differences of the media types a few words are necessary on the signal processing methods and machine learning methods listed in Table 2.1. All of these methods are described in this book, the majority in the second part. The reason is simply that the best-performing methods have a level of sophistication considerably higher than the methods discussed in the first part of the book. However, the simpler methods are required to understand the more sophisticated ones and in more than one case they

Application	*Media*	*Signal Processing*	*Machine Learning*
3D Model Retrieval	3D Model	Samples	Structural Alignment, Metrics
3D Reconstruction	Image, Video	Interest Points	Not required
Affect Recognition	Video	Templates	Mixtures
Age Estimation	Image, Video	Spectral	KNN, Metrics
Biological Categorization	Taxonomies	Not required	Association Measures
Case-Based Reasoning	Any	Any	Learning Machines
Concept Recognition	Audiovisual	Bag of Features	SVM, Mixtures
Copy Detection	Audiovisual	Color, Interest Points	Metrics
DNA Analysis	Gene strings	Samples	Association Measures
Environmental Sound Recognition	Audio	Loudness, Rhythm	Mixtures
Event Detection	Audiovisual	Time-based, Color, Optical Flow	Mixtures
Face Identification	Image, Video	Color	Metrics, KNN
Face Recognition	Image, Video	Bag of Features	Mixtures
Film Analysis	Video	Texture, Shape, Optical Flow	Metrics
Flow Detection	Video	Optical Flow	Averaging
Genre Classification	Audio	Pitch, Rhythm	Clustering, SOM
Gesture Recognition	Video	Optical Flow	Mixtures
Handwriting Recognition	Image	Interest Points	KNN, SOM
Human Action Recognition	Video	Optical Flow	Mixtures
Image Retrieval	Image	Color, Texture, Shape	Metrics, Learning Machines
Information Retrieval	Text	Terms	Bayesian
Iris Recognition	Image, Video	Spectral	Learning Machines
Language Processing	Text	Phonemes	Markov Processes
Medical Image Retrieval	Image, Video	Interest Points	Metrics
Music Retrieval	Audio	Pitch, Timbre	Markov Processes
Number Plate Recognition	Video	Interest Points, Shape	SVM, KNN
Optical Character Recognition	Image	Interest Points, Shape	SVM
P300 Detection	Time series	Peak Detection	Metrics
Panorama Stitching	Image, Video	Color	KNN
Person Identification	Video	Color, Interest Points	Markov Processes
Porn/Violence Detection	Audiovisual	Spectral, Color, Optical Flow	Learning Machines
Query by Humming	Audio	Rhythm	Learning Machines
Sex Classification	Image, Video	Templates	SVM
Speaker Identification	Audio	Rhythm, Timbre	KNN, SVM
Speech Recognition	Audio	Spectral, e.g. MFCC	Markov Processes
Stock Analysis	Time series	Autocorrelation	Metrics
Traffic Surveillance	Video	Optical Flow	KNN
Unusual Event Detection	Audiovisual	Optical Flow	Mixtures
Video Indexing	Video	Color, Optical Flow	Clustering, SOM
Video Summarization	Audiovisual	Spectral, Color, Optical Flow	KNN, Mixtures
Video Surveillance	Video	Optical Flow	Learning Machines

Table 2.1: Some Media Understanding Applications, their Media, typical Signal Processing Methods and typical Machine Learning Methods.

perform not much worse than the best method available.

The media types determine to a large degree the methods that are employed to solve a particular application problem. Figure 2.1 is an attempt to organize

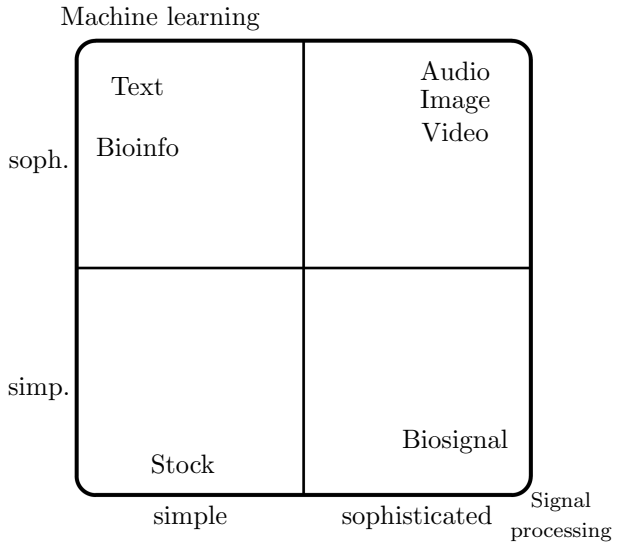


Figure 2.1: Media Types by Level of Sophistication.

the media types according to the level of sophistication of the methods that are usually employed on them. The most sophisticated methods are employed for the applications on audio, image and video data. In comparison, biosignal processing applies the majority of methods for signal summarization that are also employed on audio, but the machine learning operations used for computational understanding of the signals are rather simple. For example, in the medical domain media understanding is to a large degree left to the human specialist. Of course, no computer can read an ECG graph as good as a trained doctor but that is only true as long as the doctor looks at the ECG. Computer-based template matching and monitoring could provide an additional level of security.

Text and bioinformation share the same level of sophistication with audio, image and video in terms of machine learning. However, the signal processing employed on these data is often very basic. In the text domain methods for summarization usually work on a word-by-word basis while in bioinformation processing, for example, junk DNA is recognized by start and stop codons. It is one task of this book to argue for the transformation and application of established signal processing operations in the text and bioinformation domains. Eventually, stock data applications have seen the comparatively smallest level of sophistication. It is quite obvious that more intelligent signal processing could be performed on stock data than just sliding averages and regression. In the machine learning dimension, the arbitrariness of template matching with triangles,

butterflies, etc. should be replaced by rigorous machine learning procedures that apply objective criteria on the data. In summary, all the mentioned media types and their associated applications should be investigated by elaborate media understanding procedures. This book is an attempt to move the state of science towards the sophisticated end and to push the overall media understanding frontier a bit.

Figure 2.2 goes one step further than the last figure and puts the sets of methods employed on the different media types into context. The interpretation is the one of a Venn diagram where the axes can – roughly – be associated with signal processing (horizontal) and machine learning (vertical). Obviously, image and video understanding share the majority of methods. Video understanding extends image understanding by motion analysis that is impossible without temporal information. Both visual areas share a large proportion of methods with audio understanding. Common methods include, for example, the detection of rhythmic patterns (called textures in the visual domain) and of peaks (interest points in the visual domain). The machine learning procedures applied in both areas are mostly the same.

Biosignals are located in the figure as a subset of audio understanding. We are not aware of a fundamental method for signal processing that would be applied on biosignals but not on audio. Minor differences such as the choice of particular wavelet functions (see Chapter 12) do not count as fundamental. In terms of machine learning, audio understanding is far advanced compared to biosignal processing. Like on biosignals we consider the set of methods employed on stock data a subset of what is applied on other time series. The number of methods employed today is remarkably small considering, for example, the amount of money often involved in decisions based on technical chart analysis. Practically, we can see hardly any reason why not the majority of sophisticated audio understanding methods should be applied on this type of data as well.

Text and bioinformation play a special role in this visualization. Text understanding shares the majority of machine learning methods with sensual media understanding. However, some methods have been developed for text understanding that have not yet been successfully transferred to the other domains. Examples include boolean retrieval (see Chapter 6 and the binary independence model (see Chapter 9). Bioinformation understanding shares some methods with the text domain (e.g. the Hamming distance) but as well with audio understanding (e.g. dynamic time warping). Most of these methods are explained in Chapter 28.

Why have we chosen exactly these media types for a book on media understanding? Other media types mentioned in Table 2.1 are 3D model data and taxonomies. The first is an example for graph data while the latter are examples for structured data, a domain that also includes markup text. Another emerging data type is 3D video. Why not these data types? To start with the last entry

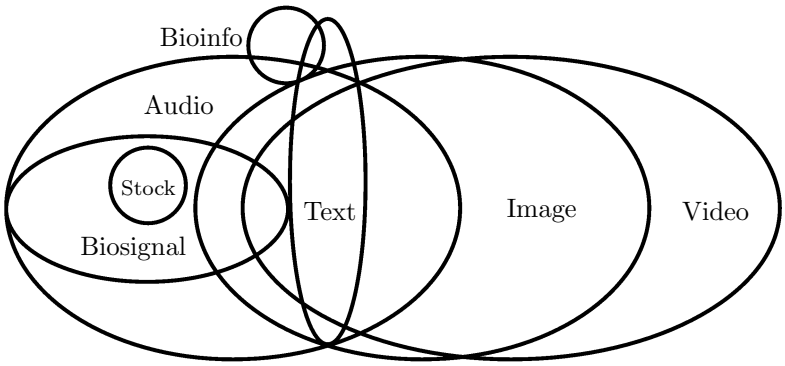


Figure 2.2: Media Types by Method Sets.

on the list, we have not included 3D video because it is not yet clear how this data type will eventually be represented. If it will be provided as two channels of video, we can deal with it in the same form as with audio, which is commonly provided with multiple channels. If 3D video is provided as one video stream with an accompanying stream of depth masks, it is crucial to have detailed information on the structure of these maps. Depth information would, obviously, be very helpful for object detection. However, due to a lack of predictability this topic has to be postponed until established standards are available.

Structured data can easily be represented by graphs. For 3D model data, the graph is the natural representation, for taxonomies it is a straightforward representation. Both data types are not investigated here because sophisticated methods are available in the graph matching domain that are fundamentally different from the signal processing and machine learning operations employed in media understanding. Deterministic and efficient graph traversal, cutting and matching procedures have been developed during the last half century. Hardly any communalities exist between these methods and the mostly fuzzy methods employed in media understanding. Therefore, these media types are not considered in this textbook. In conclusion, we believe that audio, bioinformation, biosignals, images, stocks, text and video are the most important media types for media understanding. In the next section, we investigate their most relevant properties.

2.2 Properties of Digital Media

In this section, we discuss only those properties of the media under consideration that are most relevant to media understanding applications. We employ a

comparative approach, i.e. try to work out communalities and differences in the structure of our media types. The reader should get a feeling for the scale of the quantity of information that can be embedded in the media.

Medium	*Chns*	*Dims*	*Sample Size*	*Carrier Type*	*Carrier*	*Self-Similarity*
Audio	1-6	1	2^{16}	Quantity	Amplitude	Sinusoid
Bioinfo	1	1	2^{5}	Symbol	Amino Acids	Gene Patterns
Biosignal	1-64	1	2^{16}	Quantity	Electrical Potential	Pseudo-Sinusoid
Image	1	2	2^{24}	Quantity	Color Intensity	Color Space
Stock	1-3	1	2^{14}	Quantity	Money	Pseudo-Gaussian
Text	1	1	2^{7}	Symbol	Character	Language Patterns
Video	1	3	2^{24}	Quantity	Color Intensity	Color Space

Table 2.2: Important Media Properties.

Table 2.2 lists fundamental media properties. For each media type, the normal number of channels (Chns), the number of dimensions (Dims), sample sizes and several other properties are given. In the following paragraphs, we discuss these characteristics. Most media types have exactly one channel (e.g. video). Audio may provide one (mono), two (stereo) or up to six channels (e.g. five channels and one subwoofer). The information in these channels may partially be redundant (closely located microphones) or completely independent (e.g. separation of voices and instrumentation). Biosignal recording setups usually provide twenty or more channels. The 64 channels in the table are just a typical practical hardware limit. Theoretically, hundreds of channels with individual brain, muscle, eye, heart, etc. signals could be recorded from one person. Stock data has at least one channel (open/close values) but may also provide daily maxima and minima.

Time / Space	*Yes*	*No*
Yes	Video	Audio, Bioinfo, Stock
No	Image	Bioinfo, Text

Table 2.3: Media Dimensions.

Most media types provide one-dimensional channels. For Audio, biosignals and stock data, the dimension is a time line. Bioinformation provides just a nominal scale (see Chapter 7) without a temporal context. Text is somewhere between these two types of dimensions. In words, the direction of writing hardly matters, but becomes more relevant in the grammar of sentences and very important in paragraphs and documents. Images are two dimensional without a temporal context while video provides images (frames) that are organized along a third, temporal dimension. The major difference between images and video

frames is their resolution, which is – despite high-definition television – dramatically lower in video frames. Table 2.3 summarizes the classes of media according to the presence of temporal and/or spatial dimensions.

Samples are the smallest units of media objects. In visual media, samples are pixels, i.e. points with a defined color. In audio, samples are the amplitudes of the signal which relate to the distortion of the membrane in a loudspeaker. Similarly, biosignal samples are the potentials read from electrodes attached to the body. Stock data is calculated from buying and selling operations at stock exchanges. All these types of samples are quantities (referred to as *carrier type* in the table), i.e. numbers from some range of numbers $[a, b]$. Bioinformation and text samples, in contrast, are symbols, i.e. elements from some set $\{a..z\}$.

The difference between quantities and symbols is essential for the understanding of the media types. Media built from quantities are generally redundant. That is, *neighboring* samples along one or more dimensions are correlated. The type of correlation may be a sinusoid, Gaussian, or follow the laws of a psychophysical color space. The self-similarity column in Table 2.2 names a few possibilities. In any case, transitions are somehow smooth. In contrast, media built from symbols are per se not redundant. Neighboring symbols need not be correlated at all. An extreme example would be a random sequence of symbols. Signal processing on such media would be meaningless. It is therefore relieving that media like these do not exist in practice. Even in symbolic media streams neighborhood has a meaning: for example, in words built from syllables, phrases that follow a pre-defined grammar and particular sequences of amino acids that define genes. The practical ubiquitousness of the concepts *neighborhood and correlation* in symbolic media is one justification for us to include these data types in this discussion. Correlation causes redundancy, and redundancy can be eliminated by signal processing for the benefit of better categorization by machine learning.

Another yet unmentioned element of Table 2.2 is the carrier of the samples that may range from amino acids over colors to money. The carrier is of highest significance for the sample size. Colors, for example, require the description of three-color elements (three stimuli theory, see Chapter 23 for details). Depending on the type of audio, 8-16 bits may be required for adequate digital representation. Generally, sample sizes for quantities are chosen big enough for covering all reasonably expectable values. Quite differently, symbolic carriers allow for argumentation of the sample size. For example, text need not be represented by letters. Phonemes, syllables and words would also be good carriers that would require differently sized samples. In the bioinformation domain, chromosomes could be represented on the DNA level or on the gene level. For this book, we have chosen the representations that are the most common ones in today's science.

Before we continue with a rough estimation of bandwidth requirements of the

media types, it appears beneficial to review the time dimension in the context of digital media. Such media objects are distinguished by discrete dimensions, i.e. the number of values between two points on any dimension is limited. Dimensions are, therefore, at most interval-scaled but often just ordinal-scaled. Moreover, all dimensions, e.g. the spatial dimensions of images, have practical limits. All of these properties are equally true for the time dimension. The only difference is that there is no reasonable reversion of the time dimension. While, for example, a mirrored image is still perceivable, reversed audio usually has no meaning (except diabolic messages, of course). Reversed biosignals have no meaning at all. That is, the time dimension has a natural origin at $t = 0$. In all other cases, the origins are just by definition (e.g. the lower left corner of an image). Since the natural origin of the time dimension is the only difference we could identify and this difference is merely of minor importance in media understanding we conclude to treat media types that have a time dimension like all other data types and to apply the same method on the time dimension like on all other dimensions one media type may have.

Medium	*Media Object*	*Sample Frequency*	*Size (bits)*
Audio	One Hour Audio CD	44100 Hz	2.10^{13}
Bioinfo	Human Genome	–	1.10^{11}
Biosignal	One Hour EEG	20 channels, 10 kHz	5.10^{13}
Image	Portrait Photo	600 dpi	6.10^{12}
Stock	One Year Chart	3 values per day	2.10^{7}
Text	This Book	–	3.10^{8}
Video	On Hour PAL Video	720x576 px, 25 fps	6.10^{17}

Table 2.4: Media Examples.

So far, we have avoided the problem of bandwidth requirements of different media types. The reason is that the bandwidth requirements are usually computed from sample size and sampling frequency. The latter attribute, however, is not applicable to all media types under consideration. Rather, Table 2.4 lists a few typical examples of media objects together with their size. Where relevant, the sampling frequency is given as well. For the sake of easy comparison, where possible, examples of one hour length are given. As can be seen from the table, video has by far the highest bandwidth requirements. Audio, image, biosignals and bioinformation form a relatively homogeneous group a few orders of magnitude behind video. However, observe that audio, image and biosignals use quantities as samples while bioinformation uses symbols. Since in the latter case the level of redundancy is significantly smaller (see above), the amount of information present in bioinformation is – on average – likely to be much

higher than in the two other media types. Hence, performance issues play an equally important role in bioinformation processing as in video processing. Text data and stock data trail behind the other media types. Therefore, processing should be much easier in terms of performance but this gain is paid with a lack of information. Both data types are basically free of noise (except a few typos or a false notation from time to time), but they provide only little input for sophisticated media understanding. This problem is, in particular, evident in stock data where the complex conclusions of technical chart analysis can hardly be justified by available input data. In summary, most media types under consideration have comparable bandwidth requirements. In the top group (video, bioinformation, etc.) performance issues are of high relevance due to the high bandwidth requirements.

The discussed media properties are only a few important ones. Many more do exist for the individual data types. Since we cannot cover all aspects of the seven media types under consideration here, we refer the interested reader to relevant literature. Audio, image and video properties are, for example, discussed in [298], [129], [391]. Biosignals and bioinformation are covered in [331], [225]. Eventually, text properties and stock properties are discussed in [262], [185]. Besides the named ones many other excellent sources do exist.

2.3 Media Description

This section provides the mathematical notation used for media types, media objects and operators employed on media objects. We deal with the following media types: audio, biosignal, bioinformation, image, stock, text and video. In order to be able to treat all media types in the same way we have to cover the different numbers of dimensions, the variable numbers of channels, the varying sample types and the varying sample sizes.

Throughout this textbook, all media types are represented as *arrarys* of media *samples* organized by *locations*. For example, the general media type O is defined as:

$$O = [s_l | s \in S \wedge l \in L^d] \tag{2.1}$$

where the s_l are samples from a *set* S and l is a bound location variable from set L which has d dimensions. The set S contains all numerical values and all alphanumerical symbols that may be required in one of the media types under consideration. That is, we do not define colors as three- or two-dimensional spaces of color channels but as unique numbers that identify unique colors. This is a flexible approach of gathering the various ways of color representation under one umbrella. It is sufficient for media understanding applications.

The locations set L is numeric and contains all possible indexes that may be required in any of the media types. All media types are isomorph to the general media type O. Where necessary, they are referenced as O_{name}. Please note that we do not provide a strong type system. The definition of the sample set allows for the expression of meaningless media types. We consider this shortcoming minor in comparison to the gain of understandability of the notation due to simplicity. Eventually, we have chosen the array as the data type over the – in mathematics more common – set, because it stresses the organization of the samples in the media objects. In fact, in all media types under consideration, the location (below, referred to as context) of a sample is crucial for its meaning. Sets are not per se ordered. A second advantage of the array is the straightforward implementation in computer programs. Media understanding is not a theoretical undertaking but always ends in practical implementation. Arrays are easy to implement in any imperative programming language.

The location mechanism is crucial for our media concept. To start with, the locations set L is just a container for all locations that may occur on any of the dimensions a media object may have. For all media types under consideration, the set $\mathbb{N}^+$ is a sufficient locations set. For locations relative to an origin we need $\mathbb{N}^-$ as well. The dimensionality d equals for each media type the number of dimensions given in Table 2.2. If we deal with a media object $o \in O$ we assume implicitly that the dimensionality of the location vector employed on o equals the number of dimensions given above. Therefore, locations are not per se comparable between media types but this irrelevant fact (one media understanding application deals with one media type) remains hidden in the notation. Typical locations are points L_{point} (image, video), times L_{time} (audio, biosignal, stock) and positions L_{pos} (bioinformation, text).

We would like to close this section with the definition of a few functions on media objects. The most important is the neighborhood operator $y = \theta(o, l, \epsilon)$ that cuts a neighborhood ϵ around location l from media object o and returns it as media object y (isomorph to o). Neighborhoods are highly relevant in many areas of media understanding. The neighborhood is defined by a set of locations relative to l: $\epsilon \in \{l_i\}$. Table A.2 in the appendix defines a few frequently used neighborhoods, of which L_{moore} is the most important one.

Other frequently used functions include $dims(o)$, $size(o)$ and $cut(o, l_{start}, l_{end})$. The first function returns a scalar value with the number of dimensions of the media object. The function can, likewise, be applied on location objects. The second function returns the actual size of the media object. The last function cuts a media object out of object o that starts and ends at the given locations. By this mechanism, arbitrary chunks of media information can be retrieved including the elimination of undesired dimensions. However, the *cut* function cannot interpolate or extrapolate media information. For this purpose, we use convolution over an appropriate kernel as will be explained in the second part

of the book.

In conclusion, we represent media objects as arrays of samples over locations. The notation introduced in this chapter will be extended and refined in the following chapters, in particular, the next one. For a complete overview over the mathematical apparatus which we allow ourselves for media understanding, please see Appendix A.

2.4 Media Examples

The last section of this chapter serves two purposes. First, we introduce the leading example of the entire book. Where possible, the methods employed in media understanding will be explained with the help of the leading example. Secondly, we use the leading example to sensibilize the reader on the fundamental differences of the considered media types. The results of this sensibilization will provide a natural transition to the next chapter.

Figure 2.3: A typical Business Newscast (© *CNBC*).

Our leading example is understanding the content of a typical business news-

cast. Figure 2.3 displays one frame of a EMEA CNBC newscast.[2] This type of content contains video information of the anchor person and of featured events, image information on business data, including stock charts, text information on recent events, stock values, headlines, logos, time information, faces, jingles, sometimes music, etc. In summary, this type of content covers video, audio, image, text and stock data. Furthermore, biosignal events such as a rapid ECG can be anticipated from the news contents.

We will use this type of content to illustrate the methods for signal processing and categorization discussed in the remaining chapters. In total, we use twenty keyframes from one newscast of ten minutes of CNBC television. The keyframe content ranges from anchor person shots to chart analysis shots, live discussion of recent events, a CNBC jingle and advertisements. On this material, we apply methods for video and audio understanding, image understanding, text and stock analysis. Additional material on biosignals is provided from experiments of the author with an EEG brain computer interface, ECG and pulse sensors as well as a skin resistance sensor. The sources of bioinformation are cited where used.

In the remainder of this chapter, we investigate properties of the media types related to the leading example. Figure 2.4 depicts a few waveforms as they might appear in the leading example. The horizontal dimension stands for location (image) or time (other signals). The vertical dimension depends on the type of signal: amplitudes (music, speech), values (stock), potentials (EEG, ECG) and grey level (image). All signals have been normalized (see Chapter 7) to the same range of values $[-1, 1]$, therefore, absolute magnitudes have no particular meaning.

The signals have been taken from the following sources: The image data represents the one line of Figure 2.3 that goes through the nose tip of the anchor person. ECG and EEG data are from a (hopefully) healthy but tired person – the author. The stock values given are not sampled from real data but artificially generated from a Wiener process (see Chapter 16) that looks exactly like a typical stock curve. The speech sample comes from a free German audio book where a male speaker narrates Cinderella. Eventually, the music sample has been taken from the first set of Beethoven's ninth symphony. It is therefore instrumental.

Before we dive into the analysis of the particulars of these signals a few comments have to be made. Firstly, it is important to note that an equal number of samples (400) is given for each signal in Figure 2.4. In consequence, some signals are temporally stretched in comparison to others. For example, while the stock chart shows the development of over one year, the speech sample represents only 25ms – hardly enough time to express one phoneme. The music wave represents even less time: 10ms – one brief sound of the orchestra.

[2] With friendly permission of EMEA CNBC, 10 Fleet Place, London.

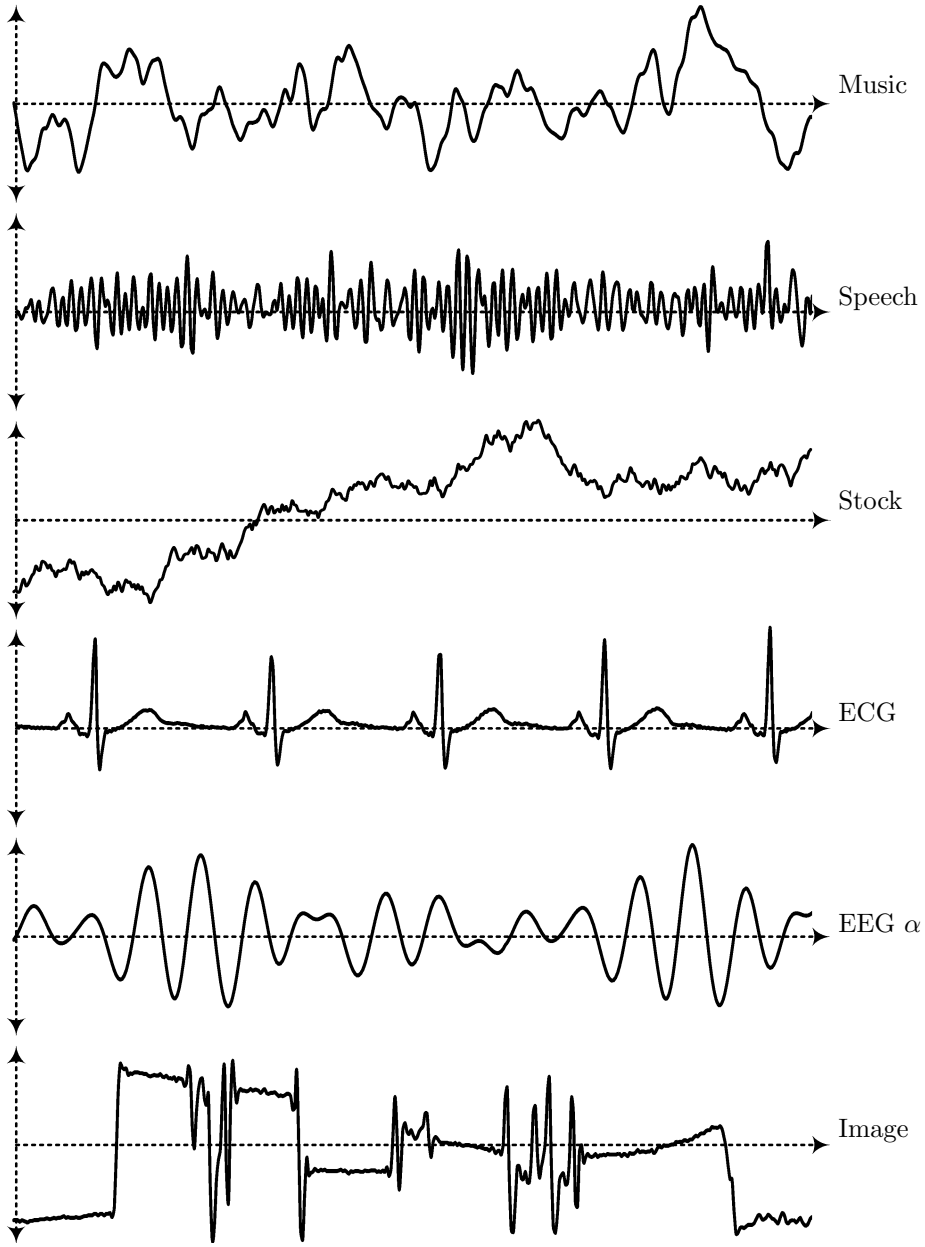


Figure 2.4: Typical Waveforms in Media Understanding.

Secondly, all the given media types use quantities as samples. We do not show bioinformation or text information. The reason is that symbolic information can only arbitrarily be organized on an interval scale (see above). In a diagram of a curve, intervals determine the difference between neighboring samples. Therefore, every curve resulting from some ordering of the symbols would be arbitrary and equally artificial/natural as any other ordering. For this reason, we excluded bioinformation and text from Figure 2.4.

The six signals visualized in the figure have communalities and differences that origin from individual characteristics. For example, the image signal is distinguished by abrupt steps in the signal. These steps (professionally, referred to as edges) represent sudden changes in the grey level. An edge appears at the border of bright and dark objects but also at line breaks. Such edges cannot be perceived in the other data types.

The ECG signal is distinguished by sudden peaks that interrupt the gentle flow of the signals. Such peaks are not present in the EEG signal. The most notable difference between an edge and a peak is that the amplitude of the signal returns to the level before the phenomenon in the case of peaks while in the case of edges, a new level is established. Another difference between ECG and image data is that the ECG signal is composed of a recurring pattern (stimulation, contraction and relaxation of the heart muscle) while the image signal has no obvious pattern.

A third notable property is the similarity of a signal to a pure sine wave. The EEG alpha wave shows a nice sinus-like function. This type of EEG signal can be captured from people that are awake but tired. If the eyes are closed the amplitude becomes bigger, if the eyes are open, the amplitude becomes smaller. In comparison, the speech signal shows a similar sine wave but with much higher frequency. The music signal is substantially more complex. The reason is the relative simplicity of the human sound creation tools in contrast to the sophisticated possibilities of an orchestra. Orchestral music has a much more complex overtone structure that destroys the sinusoid pattern.

One further, practically relevant aspect of the depicted signals is the level of noise present in each media type. Stock data will usually contain no noise while an EEG can easily contain 99% noise in contrast to 1% information. The reason is the relatively large size of the used electrodes in comparison to the small size of neurons and the high packing density of the human brain. In between these extremes, we find image data and ECG with little noise and the audio signals. For the latter the noise level depends on the quality of the recording devices and the noise environment of the recording. However, for all data types except EEG, the noise level will be rather small in comparison to the level of information.

Now, knowing the individual properties of the signals and being able to estimate to which degree they origin from the real signal and to which from noise, is it possible to express the overall similarity of the fundamental media types? The answer to this question requires the definition of signal properties (so-called *features*) and the *categorization* of the media objects – as representatives for their media types – along these features. Some reasonable features of signals could be:

- Smoothness of the signal
- Periodicity of the signal
- Fundamental frequency
- Balance of the signal

These are just examples. Many features more do exist. If we define smoothness in the sense of mathematical analysis, we can categorize (or: classify, divide, discriminate, etc.) the six signals of Figure 2.4 into three groups: speech, EEG (very smooth); music, ECG (relatively smooth); stock, image (not smooth).

Concerning the periodicity we have to judge whether or not the signals have a rhythm, a recurring pattern that may be as simple as a sine wave but also as complex as an ECG pattern. If we classify the examples by this criterion, we arrive again at three groups: ECG (highly periodic); speech, EEG (quite periodic); music, stock, image (not periodic).

The fundamental frequency is an important feature in audio and biosignal understanding. One way of measuring the fundamental frequency is to count the number of zero crossings of a signal. If we do that (naively) for our examples, we receive an ordering from stock (no fundamental frequency), image, ECG, music, EEG to speech (high fundamental frequency). However, this ordering is only partially correct. For complete correctness, the counting of the zero crossings would have to be performed on media chunks of equal length in terms of time – not samples! Then, for example, music should have a higher fundamental frequency than speech.

The last feature of the list is the balance of the signal. Let us understand a balanced signal as one that creates an equal fraction of the integral below the horizontal axis as above. This feature is closely related to the periodicity (though not completely the same). Categorized by balance our six examples form three clusters: speech, EEG (highly balanced); music, ECG (rather balanced); stock, image (not balanced).

In summary, we see that media types as different as speech and ECG, music and EEG, stock and image data share some fundamental properties. On the other side, music and speech, though both from the audio domain, are not as

similar as one would expect. Knowing these similarities and differences of media types and media objects is what matters in multimedia information retrieval. The features of a particular media set determines the choice of potentially successful signal processing operations and machine learning methods. It is one major goal of this book to convey an understanding for this connection of media properties and method selection.

The last investigation of this section is perfectly in line with this goal. We endeavor to express the overall similarity of the media types discussed above in numbers ranging from 0 (no similarity) to 100 (identity). For this purpose, we apply a so-called unitary transform (the Fourier transform, see Chapter 12) on the depicted signals, select a few of the resulting numbers as representative media properties (features) and measure their similarity as the distance between them. We employ the L_1 distance (or: Manhattan metric, city block distance) which measures the dissimilarity of two feature vectors x, y (each consisting of the N numbers created by the Fourier transform and representing one signal of the figure) as $d(x, y) = \frac{\sum_i |x_i - y_i|}{N}$. Eventually, we normalize all distance values to $[0, 100]$. Table 2.5 shows the results.

Type	*Music*	*Speech*	*Stock*	*ECG*	*EEG* α	*Visual*
Music	100	34	42	45	42	21
Speech		100	43	46	44	0
Stock			100	51	55	11
ECG				100	50	14
EEG α					100	5
Image						100

Table 2.5: Similarity of Media Types.

Of course, every signal is 100% self-similar. Further outstanding results are that speech and image are the most dissimilar types of signals. EEG comes out closest to the average of all signals. The dissimilarity to all other signals is around 50%. ECG, stock, speech and music signals follow in this order. The most characteristic signal is the image signal – maybe due to the existence of edges – which shows only a minor similarity to the music signal. This quantitative analysis can be seen as a summarization of the qualitative considerations on the four fundamental features listed above.

The major benefit of the last investigation is that it shows how multimedia information retrieval works. We start with some set of media objects that belong to the same or at least comparable media types. In the first step (signal processing), the media objects are summarized by feature transformations (here, Fourier transform and quantization to a few numbers) into feature vectors

(or: descriptions). One description represents one media object. In the second step (machine learning), the media set is organized into groups by some similarity measurement operation applied on the features (here: Manhattan metric). This general workflow of multimedia information retrieval will be formalized and discussed in the next chapter.

Chapter 3

The Big Picture of Media Understanding

Introduces the components of media understanding, sets them into context and discusses them, provides a formalism for their description and illustrates the entire process in a number of practical examples.

3.1 Introduction

In the last section of the preceding chapter we have introduced several concepts for measuring the general similarity of media objects. This model is generalized into the big picture of media understanding in this chapter. We discuss the flow of information, analyze the properties of the building blocks, formalize them mathematically and give a number of examples for multimedia information retrieval applications that follow the big picture.

So far, we used very general terms to describe the functionality of multimedia information retrieval. Signal processing is employed for the summarization of media objects. Machine learning is employed for the categorization of summaries into distinct classes that have a meaning on a semantic level sophisticated enough for human understanding. In the example in Section 2.4 we introduced more terms that describe specifically what types of signal processing and what types of machine learning are applied in media understanding. These terms are arranged into the big picture of media understanding in Figure 3.1.

In the figure, *feature extraction* stands for the signal processing component of multimedia information retrieval. This process extracts summaries from ar-

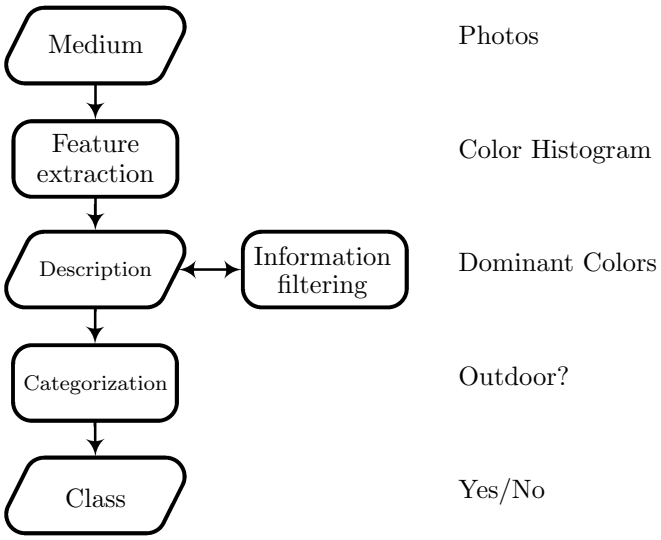


Figure 3.1: Big Picture of Media Understanding.

bitrary media content. Summaries are frequently called *descriptions* or *features*. The latter term is, in particular, common in computer vision. However, we prefer description over feature since this term has a more general meaning and – as data resulting from a transformation process – cannot be confused with the process of feature extraction. *Information filtering* is another type of signal processing that is part of the media understanding process. As depicted, information filtering executes a transformation on the description. The term *categorization* stands for the machine learning aspect of multimedia information retrieval. Categorization transforms a description into a *class* label. In most areas of research subsumed here as media understanding *classification* is more common than categorization – the latter being a term originating from psychological research of human similarity assessment. We consider categorization more general than classification. Throughout this textbook we employ the term categorization for the process of transformation of a description into a class. Individual methods are – in line with computer science practice – called *classifiers*. In summary, the two processes feature extraction and categorization transform media content into a descriptions and, eventually, a class label. Alongside, information filtering transforms the extracted descriptions into more efficient ones.

The right column of Figure 3.1 gives an example of a visual media understanding application. We presume that the given media objects are images (media type). Our goal is to categorize them into outdoor photos and indoor photos.

This end may be the first step in a taxonomic organization process. Outdoor photos could be further categorized by the season, daytime, etc. For the categorization outdoor/indoor we employ a simple feature extraction method called color histogram. This method iterates over all pixels of an image, categorizes each point into one basic color class (e.g. red, green, blue, etc.) and counts for each basic color the number of pixels in the entire image that belong to it. The result is a histogram where the colors constitute the nominal axis of independent variables and the counts of pixels are the dependent variables. Color histograms are frequently used descriptions in image and video understanding.

Before the categorization is performed, the descriptions are filtered. We select only the three dominant basic colors (those which have the most pixels associated) as representative descriptions of each image. The eventual categorization uses a simple decision tree as classifier. If green and blue are among the three dominant colors we consider an image as outdoor and associate a class label '1' otherwise '0'.

The reasoning behind this example is that green is the typical color of plants while the sky is blue. Both of them should be visible outdoors but not indoors. However, the experienced reader will be aware that this model will perform well often – but not always. In practice, situations not foreseen by the experimenter emerge that require adaptation and refinement of the employed feature extraction and categorization methods. However, the big picture remains the same. *Media retrieval is an iterative process of feature extraction and categorization.*

Below, we investigate the information compression aspect of the big picture as well as, in the next section, all steps and results of the process in detail. For now, it is worth noting that this simple model is sufficiently general to describe the concept of media understanding independent of the type of media and the type of application. Together with the general mathematical formalism developed for media representation (last chapter) and processing steps (this chapter) the big picture provides a flexible tool for the resolution of multimedia information retrieval problems.

Due to its importance we use the big picture also as the organizing principle of the three parts of this book. Each part starts with chapters on media properties and/or feature extraction. Then follows one chapter on information filtering in each part. The remaining chapters of each part deal with categorization problems and solutions.

The big picture is a generalization of the multimedia information retrieval process. As most generalizations it is also a simplification. In the remaining chapters of the first part the simple model is sufficient to set all described methods into context. In Chapter 11, however, we will extend the big picture by properties essential to sophisticated media understanding. Until then, we consider only the following three aspects too important for being neglected in the simple model.

- Querying situation
- Feedback loops
- Information filtering

The typical querying situation in media understanding comprises a *query* and a *media database*. The query may be an element of the media database or not. In the latter case it is usually provided by the user. Essentially, the query may be one media object, a group of media objects (so-called *query by example* approach), a coarse representation of a media object (*query by sketch*) or some form of description (for example, *query by text* in text retrieval). Single querying objects are usually considered positive examples for the query. If a group is provided it is common to label each element of the group as a *positive* or *negative example* of the query. Here, positive example means that the multimedia information retrieval application should identify media objects *similar* to the query object. The application of groups of query objects can be done in various ways that are described in the categorization chapters of this book.

The typical flow of querying requires that in the first step, feature extraction has to be performed on the query object(s). The feature extraction on the media objects in the database can be performed offline. The general performance problem of media understanding (see Chapter 1) is, therefore, mostly a categorization – not a feature extraction problem. In the second step categorization is performed in one of two ways: The first option is to derive the class label of a query object from the description and to select media objects from the database that belong to the same or a similar class. The second option is to compare the descriptions gained from the query object(s) to those of the media database. Matches can be based on global (entire descriptions similar) or local (parts of descriptions similar) level. Eventually, a *result set* with at least one most similar match is presented to the user. Based on the result set the multimedia information retrieval process can be refined by consecutive queries.

The last sentence already implies the existence of feedback loops in media understanding. In fact, feedback is of highest importance for the querying process. Feedback may be provided by the user or the implementer of the media understanding system or by the system itself. If provided by the user, feedback is usually given in the form 'These objects in the result set suits the query. These do not.' Such *relevance feedback* is employed for iterative refinement of the query. If feedback is provided by the experimenter then it is usually called *ground truth*, i.e. a set of media objects associated with human-rated class labels. Ground truth is of highest significance in classifier training, as we will see in the first part of the book.

Eventually, the system itself may provide feedback on many levels. In fact, typical media understanding is an iterative process of media understanding cy-

cles. In the example above we employed a first media understanding cycle for the categorization of individual pixels. These classes were exploited for a second iteration in which we categorized media objects by dominant colors. This is a simple example. In state-of-the-art media understanding applications many more feedback loops are utilized. Media understanding may first be applied on individual media channels of media objects (e.g. audio and frames of video objects), divided into chunks of data, aggregated temporally and spatially and so on.

One particularly important feedback loop is the initial transgression from *quantitative* to *qualitative categorization*. Quantitative categorization is the almost inevitable first step of media understanding. Descriptions derived from media content are mostly quantities – like the samples of quantitative media types. In the first iteration of categorization these quantities are transformed into class labels, i.e. symbols. In the example above, we transformed dominant colors (quantities) into an outdoor/indoor (qualities) categorization. Further media understanding iterations could employ the qualitative descriptions in order to categorize outdoor photos, as mentioned above, by season and daytime. Using class labels as descriptions for the purpose of increasing the interpretability of the output is called *semantic enrichment* in media understanding. Obviously, the transgression from quantitative to qualitative categorization is irrelevant for symbolic media types such as text and bioinformation but, of course, media understanding on these data types may have feedback loops as well.

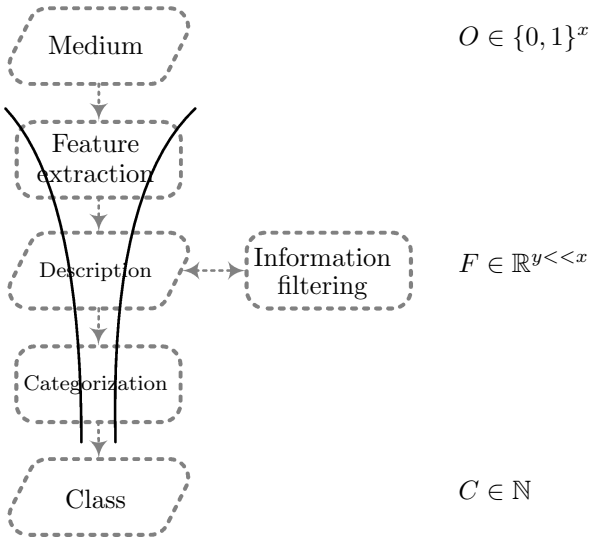


Figure 3.2: Information Filtering by Media Understanding.

The last aspect of the big picture to be mentioned here is the information filtering effect of feature transformation and categorization. Figure 3.2 illustrates the effect. Media objects are large blocks of binary data. In comparison, descriptions are – in the worst case – vectors of real numbers with significantly lower dimensionality. Real numbers are, in fact, only required for quantitative descriptions. Classes can be expressed in $\mathbb{N}$. Hence, the qualitative description reduces the amount of information even further while enriching the semantic meaning. Eventually, media understanding results in one class label per media object (in a simple case, 'similar to the query' or 'not similar'). That is, the large block of media content is reduced to one number. However, this number is semantically highly loaded. The class label is only meaningful with respect to the given query. In conclusion, media understanding is an iterative process of semantic enrichment by information filtering and categorization that transforms the general content of a media object into a specific answer meaningful only to a particular query.

3.2 Elements of Media Understanding

In the last section of this chapter we will give more examples of particular media understanding applications that are based on the big picture. This section, however, is dedicated to the *elements* of the big picture. We discuss the terms of Figure 3.1 generally and with respect to the fundamental media understanding problems listed in Chapter 1.

Feature transformations reduce the media content to uniform descriptions. From visual material, color information, texture information and the shapes of objects can be extracted. From video, additionally, motion can be extracted. Text is typically reduced to the principal parts of the most relevant words. From biosignals and audio energy, peaks and rhythms can be extracted. Eventually, from bioinformation, for example, fundamental genes can be extracted. A good feature transformation will try to anticipate the categorization process. That is, the descriptions of media objects belonging to the same class will be very similar while the descriptions of media objects belonging to different classes will be significantly different. Such a feature is called *discriminative*.

We will discuss the qualities of good feature transformations in more detail in later chapters. However, Figures 3.3 and 3.4 illustrate the difference between discriminative feature transformations and not discriminative ones. In the latter case, the descriptions of a given media database are uniformly distributed over *description space*. In the first case, however, areas with higher density (clusters) are separated from areas with lower density. Discrimination requires the existence of such a structure as a necessary condition. However, the structure is not a sufficient condition in all cases. If members of different classes were

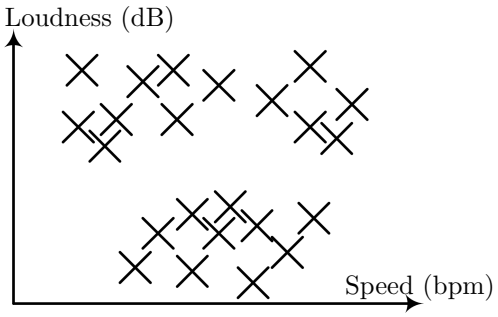


Figure 3.3: Description of Musical Objects by two good Feature Transformations.

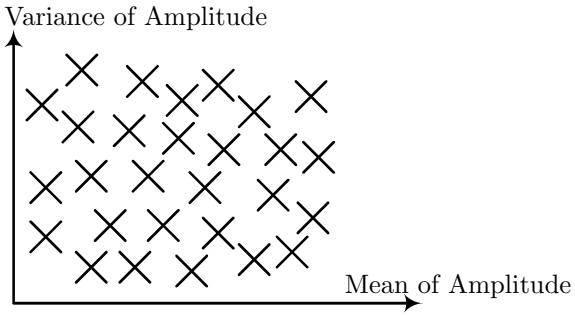


Figure 3.4: Description of Musical Objects by two bad Feature Transformations.

represented to the same extent in all clusters, the descriptions would be as non-discriminative as in the case of Figure 3.4 – but this is in practice seldom the case.

The implementation of feature transformations touches the fundamental problems of polysemy, curse of dimensionality and handling of noise, distortions and missing data. Polysemy is actually reduced or eliminated by the feature transformation process because feature transformation is an interpretation step that transforms a perceivable media event into an abstract numeric representation. The abstract representation does not allow sensual perception and, therefore, reduces the possibility of misinterpretation to the statistical level of membership in one or another densely populated area of data points. The curse of dimensionality, on the other hand, is mainly created by the feature transformation process. As we will see in the forthcoming chapters, feature transformations employ recurring building blocks with one or a few parameters each and a few reasonable values per parameter. If a handful of building blocks are merged into

one feature transformation, the number of possible parameterizations grows exponentially. Calling this property a curse is very appropriate. A lot of work of feature transformation design flows into the elimination (e.g. by constraints) of extrinsic parameters. Eventually, noise, distortions and missing data are problematic because if not handled intelligently by the feature transformation, these unwanted signal components would be transformed into legitimate description components. Since most descriptions are abstract, noise can only be detected statistically and, therefore, within hardly satisfactory limits. In summary, these three fundamental problems influence the feature transformation process and have to be considered in the media understanding design process.

The feature transformation process creates a description, i.e. a vector of numbers. If extracted from quantitative samples these numbers will be *real values*. If extracted from symbolic samples or iteratively from class labels, the numbers will be cardinal numbers or even binary predicates. The vector structure is important for categorization. In the heart, most categorization methods require the comparison of pairs of descriptions. One dimension in the description vector represents one property of the media object. The vector format is very convenient in order to guarantee that in the categorization process dimensions of one description are always compared to the appropriate dimension of the other description. It is, therefore, common to transform matrix data and multi-dimensional output of feature transformations to a vector format. Vector descriptions created by multiple feature transformations can simply be merged by concatenation. However, it is important to note that descriptions have to be made independent of media size in order to guarantee comparability. This is often achieved by estimating the maximal descriptor size and filling up positions unused by smaller media with zero values. Details of description merging and normalization will be discussed in Chapter 7.

Categorization transforms description vectors into class labels. Depending on the querying principle the categorization process is either performed example-based or rule-based. In the first case, the *classifier* derives the class label of some media object by its similarity to examples of the classes. In the second case, the classifier employs some inference rule on the description elements. These two fundamental approaches are not as different as they may seem. The rule-based approach requires a learning step prior to application that usually implements a functionality very similar to the example-based approach. In the chapters on categorization we will endeavor to introduce the majority of categorization principles employed in machine learning. Reviewing these principles in the second part of the book will show that only a handful of principles is recombined in the – at first sight highly different – approaches.

The core functionality of a classifier is to group similar descriptions and associate groups with a class label. The quality of the grouping stands and falls with the quality of the descriptions provided by feature extraction. Properly defined

assumptions can, to some degree, reverse the negative effect of a bad feature transformation. However, no classifier is able to classify a space of uniformly distributed descriptions.

Actually, no classifier *should* be able to categorize such a space correctly. A classifier that can be trained to discriminate even such a space more or less correctly into classes would be called *overfitting*. That is, its learning procedure would adapt too far to the training data. This statement may appear surprising. Why should precise adaption to the training data be a problem? The problem is that no machine learning method employed for categorization can be trained on all descriptions that may appear in practical application. Due to constraints in time and availability of training data the training is always limited to a subsample of the space of possible media objects. That implies, statistically, the danger that the distribution of descriptions in the training data is not the same as in the, unknown, media space. A classifier that adapts perfectly to the characteristics of training data that does not represent the media space correctly would, therefore, fail in practical application. Since the problem of inappropriate representation of media space by the training data exists in most media understanding applications, overfitting has to be cured by a proper scheme.

This scheme is the implementation of a desired level of inflexibility of the classifier (frequently, referred to as *structural risk minimization*, see Chapter 18). Overfitting can be avoided if limiting constraints are added to the model of the classifier. In consequence, the training data can only be learnt to a certain degree. The benefit of this approach is that such a classifier is less prone to overfitting. On the other hand, the classifier is not able to learn a sophisticated class structure perfectly. The practical implementation of categorization methods is very much about *identifying the optimal trade-off between model flexibility and overfitting avoidance*. This trade-off has to be identified for every media understanding application anew. It is, therefore, a goal of this book to focus in the description of categorization methods on similarities and differences of the existing approaches in order to make the reader understand which method should be applied when.

Categorization has to deal with the following fundamental media understanding problems: sematic gap, incomplete ground truth and performance. The second problem refers to the before-mentioned problem of selecting an appropriate sample for training. The solution has been sketched in the last paragraph. The semantic gap problem [322] is practically of highest importance. A media understanding application that is not able to capture the semantic concepts provided by a non-expert user will cause frustration. In consequence, it will not be used. There is currently no full remedy against the semantic gap problem. The most promising approach is to use as many feedback loops as possible, re-use the categorizations learnt in earlier iterations and to *put the human in the loop*, i.e. use semantic feedback of the user (e.g. 'this one is good') for refining the fea-

ture transformation and categorization process. Throughout this book we will emphasize promising methods for reducing the semantic gap problem.

Above we judged the performance problem as of only minor relevance to the feature extraction process. Our argument was that, depending on the querying paradigm, only one or a few examples have to be transformed into descriptions at runtime. All other objects of the media database can be transformed offline. This argument is not valid for the categorization process. The categorization rule has to be as simple as possible since it has to be employed at runtime on any media object in the database (exceptions neglected for the sake of simplicity). However, a simple categorization process will not be as successful as a complex one. This is another trade-off problem. Depending on the media understanding problem an appropriate categorization method has to be designed that is fast enough for quick response times and sophisticated enough for minimizing the semantic gap problem. In practice, a frequently used approach is to employ a sophisticated training process that can be performed offline and that results in a rapidly executable decision rule for online application. As we will see in the forthcoming chapters, the most successful machine learning techniques employed in media understanding follow this principle.

Eventually, the categorization process produces a class label. It is common to label classes numerically but, of course, these numbers may stand for arbitrarily complex semantic concepts. As mentioned a couple of times already, it is beneficial to feed the class labels of one iteration of media understanding back into the process in order to climb higher on the semantic ladder and close the semantic gap.

3.3 Description of Elements

This section continues the work started in Section 2.3. We present and discuss the formal representation of descriptions and class labels. Furthermore, we introduce a number of functions relevant for the manipulation of descriptions and class labels.

The mathematical description of descriptions and classes pursues three goals:

1. Generalization of the containers used for media objects, descriptions and classes.

2. Provision of a uniform model that can be used by all feature extraction methods and all categorization methods.

3. Comparability of methods between media types.

The third requirement is guaranteed below by not distinguishing between media types. We use the same syntax – introduced in the last chapter – for

all types of media. In order to meet the two other requirements we define descriptions as arrays isomorph to media objects:

$$F = [s_l | s \in S \wedge l \in L] \tag{3.1}$$

The only difference to the definition of O in the previous chapter is the locations set L. For descriptions F the locations set is always one-dimensional, i.e. each description is a vector of samples s_l drawn from a set S. Typically, $s \in \mathbb{R}$.

Since $F \sim O$, this definition of descriptions satisfies both requirements. The array approach serves as a generalization of media objects and descriptions. In fact, for most media types considered in this book, even the dimensionality of media objects and descriptions is the same (though not the object size). Secondly, since feature transformations take their input from media content or other descriptions we cannot imagine a feature transformation that would not be able to output descriptions of the same Gestalt as media objects.

Classes are defined as follows:

$$C = [s_l | s \in S \wedge l \in \emptyset] \tag{3.2}$$

That is, a class C is just another array but the locations set is empty. Therefore, a class is a scalar value drawn from S. In the simplest case, $S \in \{0, 1\}$ (binary classification, membership) or $S \in \{-1, 1\}$ (two disjoint classes). Practically, $s \in \mathbb{N}^+$.

This definition of class labels fulfills the above requirements trivially, since $C \sim F$. Additionally, it allows to use class labels as descriptions and feed them back into the media understanding process. We agree that the notation puts the entire formalism to an extreme, because eventually $C \sim O$, i.e. class labels are considered similar to media objects disregarding the different sizes. However, from the practical point of view $C \sim O$ is a relatively weak statement. Many research results suggest that human beings label sensual stimuli quickly and base their reasoning on the class labels instead of the actual stimuli. Following this line of argumentation would mean to state equivalence between media objects and class labels. We do not intend to go that far. For the purpose of this book, $O \sim F \sim C$ is a convenient result of the formalization process.

In the remainder of this section we introduce important functions on media, objects, descriptions and class labels. To begin with, the following functions compute statistical moments ($f \in F$): $x = min(f), x = max(f), x = mean(f), x = median(f), x = mode(f), x = span(f), x = stddev(f), x = var(f), x = skew(f)$. From the usage of description f we can see that this variable type can be interpreted as a distribution as well. Actually, most times we will not use these functions but use the variables from Section A.4 instead.

One important function that we will use frequently is the merging operator: $x, y \in F : x+y = [s_l | s \in S_x \cup S_y \wedge l \in L_x \cup \text{offset}(L_y, \max(L_x))]$ where $\text{offset}(a, b)$ adds value b to all members of set a. That is, the resulting set contains all values of y concatenated after the values of x. The merging operator can join descriptions but likewise media objects and class labels.

The convolution operator is another frequently used function in media understanding. Below, we employ $x \otimes y$ for the convolution based on the dot product: $\sum_i x_i y_i$. This operator is a similarity measure. The maximum is reached if the convolution set y is identical to the input data x. This form of the convolution operator is typically employed in image understanding. In audio understanding on the other hand, the convolution operator is frequently based on the L_1 metric used the last chapter: $x \bar{\otimes} y = \sum_i |x_i - y_i|$. The L_1 metric is a distance measure. The similarity of x, y is maximal if the convolution approaches zero. Throughout this book, we use the symbol $\bar{\otimes}$ to distinguish this negative correlation from the positive correlation $\otimes$.

The third class of functions to be mentioned here are the similarity and distance measures – a generalization of the convolution operators. We denote $m(x, y)$ for similarity measures and $m^{-1}(x, y)$ for distance measures. The output of $m(x, y)$ is maximal, if two objects x, y are identical. At the same time the output of $m^{-1}(x, y) = 0$. Hence, similarity and distance measures are defined on $[0, 1]$ with reversed meaning. As we will learn in the subsequent chapters, distance is not the direct inverse of similarity. In fact, the correlation is based on the natural logarithm. The symbol m^{-1} should therefore not be understood as inversion in a strict mathematical sense.

We would like to close this section with a brief discussion of the mother functions of the media understanding processing steps. Below, all feature transformations will be derived from a function $f = \text{transform}(o), o \in O, f \in F$. This function takes a media object o as input and generates a description f. However, since $O \sim F$ the usage $x = \text{transform}(f)$ is also valid and means that the description x is extracted from f.

Information filtering is abstracted in mother function $y = \text{filter}(x)$ where $x, y \in F$. That is, the filtering function generates a description from another description, typically by removing noise and/or redundancy. Obviously, the filtering function has the same signature as the transformation function and, actually, information filtering is just another feature transformation process.

Eventually, all classifiers are derived from $c = \text{classify}(f), f \in F, c \in C$. The categorization process transforms a description f into a class label c. However, the same is true for the categorization function as for the other two. It is just another transformation function. As we will see below this view is actually correct. Most categorization functions employ the same *building blocks* – in partially different order, but sometimes in the same – as feature transformation functions. Throughout our journey through the world of media understanding

we will endeavor to identify such building blocks.

3.4 Application Examples

The remainder of this chapter is dedicated to examples. We start with introducing the pseudo code used in the book by formalizing the example given in Figure 3.1. Then follows one operationalzation of the big picture per media type staring with text understanding and ending with bioinformation analysis.

The following algorithm expresses the media understanding example given in Figure 3.1. Here, X is a given set of media objects.

```
foreach x in X do
   y := color_hist(x)
   y := dominant_colors(y)
   z := classify_colors(y)
   print x,z
endfor

function classify_colors takes x begin
   y:=0
   foreach color_bin in x do
      if color_bin = GREEN then
         y:=y+1
      elsif color_bin = BLUE then
         y:=y+1
      endif
   endfor

   if y>=2 then
      return 1
   else
      return 0
   endif
end
```

For all elements of the media database X we extract a color histogram (derived from the transformation mother function) and do filtering by dominant colors (derived from the filtering mother function). The function *color_hist* is described in Section 5.2. The resulting descriptions y is a vector of three dominant colors. The categorization function derives a class membership for each media object. Its functionality is defined in function *classify_colors*. We count the number of color bins (elements of input x) that are either green or blue.

Since both colors must occur, an object is only classified as 1 (standing for 'outdoor') if the counter $y \geq 2$. Please note that this pseudo code format is used throughout the book. See the appendix for a list of reserved words.

The remaining examples of this section are for illustrating the flexibility of the big picture. We employ it to solve one problem per media domain by a simple algorithm. The solutions have not been chosen because they would represent the state-of-the-art but because they are – to our experience – easy to comprehend. For every example we explain the problem first. Then, we suggest a solution: a feature transformation method and a categorization method, information filtering where required. The suggested methods are not explained in detail – this will be done in later chapters. Comments will be made where fundamental problems of media understanding are touched.

We start with a text retrieval problem, because in text retrieval feature extraction is straightforward and intuitive. Categorization can be performed very effectively. Imagine a data pool of recent business news in which we want to identify the number of messages that express a positive development of the stock price of IBM. For the sake of simplicity we operationalize this problem as identifying the terms 'IBM' and 'up ... points' in the text messages. We classify the stock development of IBM as positive if 30% or more of the news that contain 'IBM' also contain the second term.

In order to solve this problem we suggest the following feature transformation on the business news items:

1. Remove all non-text content (e.g. markup) from the message.
2. Replace all tokens 'increase(s) ... points' with 'up ... points'. That is some kind of reduction to the principal parts of the terms.
3. Split the text in sentences and do the following operations for each sentence.[1]
   (a) Remove all sentences that do not contain both 'IBM' and 'up ... points'.
   (b) For each remaining sentence count the number of words between the two terms. If the count is greater than 15 use $count = 15$.
4. Fill the five lowest counts into a description vector. If less than five sentences could be identified fill the remaining positions with '15'.

The resulting descriptions are categorized as follows: Compute mean and variance of all elements of the description vector. If the mean is below, say, five

[1] Please observe that this proceeding is a typical example of iterative media understanding. In the subroutine we, again, apply a feature transformation and categorization. The result is employed in the outer media understanding process.

and variance is below four we decide to believe that this particular news item expresses a positive development of the stock price of IBM. Eventually, we only have to sum up all positively classified items, divide by the total number of news items and compare to the given *threshold* of 30%.

Many things could be said on the proposed solutions. First of all it appears completely arbitrary. Hundreds of other reasonable paths could be chosen. Please note that this is typical for media understanding. There are actually thousands not hundreds of possible paths to the solution. It is therefore recommendable to try just one that appears reasonable and, if it works, use it. If it does not work, adjust the parameters and if that does not improve results try a redesign.

Secondly, why do we compute this particular description? It has several advantages. First, it considers the amount of text between the interesting terms. The more words, the less we *believe* in the message. Second, we do not rely on just one sentence but allow up to five. This strategy should make the algorithm robust against noise. For example, if a negative news item on IBM is introduced with last week's positive performance, this would result in only one positive entry in the description – too little for positive categorization.

Thirdly, why exactly these thresholds? For no particular reason. We have just chosen them for the example. In the practical application of media understanding, trial and error is a common method. We start with some possibly reasonable thresholds. If the system works, fine, if not, we adjust them in order to meet the requirements of the media domain. The given values are just initial guesses.

Many other aspects of the solutions could be discussed here but we consider it beneficial to postpone them to the examples below and – the majority – to the remaining chapters of the book. The three problems just discussed – degrees of freedom, modeling of belief, and setting of thresholds – are of highest importance since they reappear in almost all media understanding problems.

Figure 3.5: Face Identification Example (© *CNBC*).

For the second example we chose the visual domain and try to identify faces in news broadcasts. Figure 3.5 illustrates an example. In video frames like the

one on the left side we want to identify all regions that could be faces. The application could, for example, be video indexing. If all faces in all shots could be identified reliably, the link between faces and shots could be used for quick selection of all contributions of one particular person. Please note that the goal of this application is *face identification*, not *face recognition*.

We suggest the following approach of feature transformation and categorization as solution. Each frame is analyzed individually in the following way:

1. Reduce the frame size to 64x64 pixels by averaging neighboring pixels

2. Identify all pixels that have the color of one the following three *skin color models*: African (brownish), Asian (yellowish), European (pinkish). Label all pixels of skin color as '1' and all other as '0'.

3. Investigate neighboring pixel groups labeled as '1'. If the border is more or less a circle, assume a face, if not, assume a non-face-object.

The resulting faces could be described by center point and diameter of the face region. This algorithm mixes feature transformation and categorization on various levels. The first operation provides a description of the video frame. The second is an act of categorization. The third includes feature transformation (identification of the border line, a so-called *edge*) and categorization of this line as circular/otherwise. The details of this categorization (degree of freedom of circularity) has been omitted for the sake of simplicity. It has to be mentioned that skin color models are actually very successful in face identification.

The goal of the second example is to show that the big picture of media understanding is iteratively applied in sophisticated applications. The goal is reduction of the semantic gap to a minimum. If we used just the skin color model for face identification the number of *false positives*, i.e. the number of identified regions that are actually not faces, would be much larger than in the suggested model. This model, however, could be extended by additional tests (e.g. identification of the nose tip) resulting in further reduction of the semantic gap paid by higher dimensionality of the parametrization problem and lower performance. This connection is important: higher semantic meaning has to be traded against performance in media understanding.

In the third problem we transfer the setting of the first example into the speech domain. This time we try to identify the spoken English words 'IBM' and 'up/positive/good' in the audio channel of newscasts. The application could be a preprocessing step for the first example, i.e. the provision of the text messages investigated there. However, in order to keep the problem as simple as possible, we do not suggest a solution for general *speech recognition* but only for these particular four words.

Our solution is based on a few assumptions. Firstly, we use the psychoacoustic knowledge that humans transmit speech hardly ever above 4000Hz (see Chapter 4). Secondly, we use the hint of Figure 2.4 that the pronunciation of Germanic languages results in sine waves without large changes in amplitude. Therefore, we focus our feature transformation on the recognition of different frequencies. We suggest the following algorithm:

1. Downsampling of the audio channel to 8000Hz[2]
2. Split the resulting audio stream in chunks of 30ms length. English phonemes are seldom shorter than 30ms.
3. For every chunk count the number of pairs of samples of which one has a positive sign and the other has a negative sign (so-called zero crossings).
4. Create one description vector per second of spoken audio that contains the number of zero crossings per 30ms chunk.

For the categorization of description vectors we require *references*: in the simplest case the queried words spoken by some person (often, the experimenter). References provided, we suggest the following categorization method for each of the four interesting words and each description vector.

1. Identify the item of the description vector most similar to the first element of the reference by L_1 distance. If the distance is below a certain threshold, we believe that this position in the description vector is where the reference word starts. Otherwise, the description vector is discarded.
2. If the start position has been identified, we identify the position of the last phoneme of the reference in the tail of the description vector in the same manner.
3. If the lengths of reference and start/end in the description vector are different, the latter object is scaled down by intrapolation.
4. Eventual categorization is performed by counting the L_1 distances of all corresponding phonemes of the reference and the downsampled start/end segment of the description. If the sum of distances is below a certain threshold, we believe in positive recognition of the word represented by the reference vector.

This algorithm is a simple example of *dynamic time warping* – a method used in audio understanding as well as bioinformation processing. Like the other

[2]If you do not understand why exactly 8000Hz, please google 'Nyquist Shannon law.'

algorithms above this one relies heavily on thresholds and the arbitrary selection of feature transformation methods and similarity measures. However, one major difference is that here – though we investigate a semantically complex problem – the big picture is not applied iteratively but in a straightforward fashion. One flow of feature transformation and categorization provides a (practically, not too bad) solution.

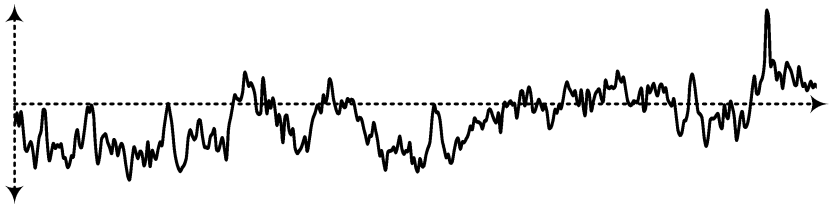


Figure 3.6: EEG β Wave.

The fourth example has been taken from the biosignal domain. In a large-scale database we want to distinguish EEG samples of people with high brain activity from EEG samples of idle people. The application could be psychological testing of stress responses triggered by falling stock prices. In Figure 2.4 we have already seen a typical EEG alpha wave (idle). In comparison, Figure 3.6 shows one EEG channel of high brain activity (so-called beta wave).[3] We can see two fundamental differences. Firstly, the alpha wave is smooth while the beta wave is not. Secondly, the alpha wave is sinusoid, the beta wave not.

Still under the impression of the last example and recalling the message of the first chapter that audio and biosignals have a very similar nature, an immediate guess for a good feature transformation applicable on this problem would be the zero crossings rate, which should be much higher for EEG alpha waves. However, if we investigate Figure 3.6 closely, we can see that, actually, this signal also frequently crosses the zero line. The difference is not in the number of zero crossings but in characteristics of their frequency. In the EEG alpha wave the zero crossings occur at intervals with small variance, in the EEG beta wave the variance of the intervals is high.

In consequence, we suggest the following feature transformation:

1. Identify all zero crossings in the signals under investigation.
2. Compute the intervals between neighboring zero crossings.
3. Compute mean and variance of interval size and use these moments as description vector.

[3]For the expert: taken from position C4 of a 10-20 mask.

Please note that the third step is actually an information filtering operation. We could use the zero crossings intervals in normalized form as descriptions as well. However, the computation of the two first statistical moments reduces the size of the description significantly. For the categorization process the mean is of little significance. If the variance is below a certain threshold we assume an alpha wave otherwise a beta wave.

The biosignal example should convey two important messages. First, whatever works on audio is likely to work on biosignals as well. We will exploit this insight in the next chapter. Second, information filtering operations can simplify a media understanding problem dramatically. It lies in the nature of information filtering that the separation of such operations from feature transformation depends on the point of view of the experimenter.

a) no b) no c) yes d) no

Figure 3.7: Newscast Shot Sequence (© *CNBC*).

The fifth example has been taken from the video domain. In the transmission of a news channel we want to identify commercials. The application could be to switch channels during commercials. Figure 3.7 shows four typical video frames together with *ground truth* expressing whether or not these frames belong to commercials.

We suggest the following solution for this problem:

1. Identify shot breaks using the following algorithm:
   (a) Convert all frames from color to gray.
   (b) Build a luminance histogram for each 25th frame that counts the number of pixels belonging to bins of 0%, 10%, etc. gray level.
2. Identify shot breaks by measuring the L_1 distance for pairs of neighboring luminance histograms. If the distance is above a certain threshold we assume a major change of content, i.e. a shot break.
3. Identify all faces in the first frame of each shot using the algorithm introduced above.

4. If no face of a well-known anchor person of this channel is visible in the first frame, we assume the start of a commercial. The end is assumed where a well-known face becomes visible again.

Certainly, this algorithm for the identification of ads is not suitable for practical application. Shots containing no faces at all would generally be classified as commercials. Practically, chances of success would be higher if we investigated the visibility of the headline instead of faces (or both). The purpose of this toy solution is a different one. Often, it is good engineering to assemble a media understanding solution from existing well-evaluated building blocks. Rapid prototyping and opportunistic error fixing are fundamental approaches in media understanding application design. Following this argumentation, using face identification would be worth trying. All occasions where it fails could be fixed by adding a second method to the categorization process.

The last example has been taken from the bioinformation domain. We want to build a media understanding application that measures the overall similarity of two DNA strings. A typical application would be the measurement of overall similarity between species. Two DNA examples could look as follows:[4]

```
GTATAAGTTC TTCTATATAG TCAATTAAAG CAGGATGCCT ATTAATGGGA AGTGTGAAAG
GACCAAGTAA GAAAAGGTTA GTAGATTTTT CAAATAAGAG TAATGTCAAT CTAGTGGTTT
```

Each string is composed of symbols representing the four nucleotides that build up the amino acids: Adenine, Cytosine, Guanine and Thymine. Each triplet of nucleotides defines one amino acid. The 64 possible combinations result in only twenty amino acids with different properties. The DNA of living organisms consists of between 2.10^5 and 2.10^{11} amino acids. Of this number, more than 90% (in the human genome, 97%) is junk DNA that does not belong to genes. Genes are mostly identified by the start codon 'ATG' and the first occurrence of one of the stop codons 'TAA', 'TAG' or 'TGA.' In order to do a general comparison of two DNA strings we need to extract the genes and align them pairwise.

We suggest the following feature transformation procedure as part of the solution. For each pair of DNA strings do the following operations:

1. Extract all genes using start and stop codons.
2. Replace all triplets of nucleotides in all genes by their number in the genetic code ($\{1..64\}$).
3. Take an offset of 32 from all numbers.
4. Count the number of zero crossings in all genes.

[4] GenBank Sample Record Saccharomyces cerevisiae.

5. Represent each gene by mean and variance of the intervals between zero crossings.

The categorization could be performed as follows:

1. Align those genes of the two strings that have the minimum L_1 distance regarding their description vectors (both mean and variance).
2. Compute a score that rewards linear alignment, i.e. if the first gene of one string is aligned with the first of the second, and so on (e.g. by evaluating the distances in position created by the alignment).
3. Categorize two strings as similar, if their score is below a certain threshold, otherwise as non-similar.

This procedure is very different from what is actually used in *DNA sequence alignment*. However, even though this is just a toy example, its practical performance is not too bad while having a significantly better performance than the standard approach of dynamic time warping of nucleotide strings.

These examples have illustrated that the big picture of media understanding is applicable in all six considered media domains. Sometimes a straightforward application is sufficient, at other times iterative application is required on multiple levels. The experimenter has to decide what is required in order to reach a minimal semantic gap at acceptable algorithmic complexity and performance.

It has to be noted that all of these examples are single-media examples. Of course, single-media understanding can be embedded in a true multimedia understanding process. Typically, feature transformation and categorization is performed on the individual media objects first. Then, descriptions are merged and a categorization process is performed that takes all individual results into account. The question of multimedia understanding will be discussed in detail in Chapter 7.

In conclusion, we would like to stress again that the presented solutions where chosen for understandability not because they would represent the state-of-the-art in media understanding. We generally believe that the best approach to media understanding is based on a comprehensive ground truth and a workbench of feature transformations and classifiers that are recombined until the ground truth has been imitated to an acceptable degree. This approach can fairly be called *bottom-up media understanding* while the examples above where analyzed *top-down*. In practice, top-down solutions hardly ever meet the expectations because the experimenter is usually not aware of all cases and exceptions that may occur in a particular media body. Therefore, we prefer the bottom-up approach. In the next chapters we will collect the tools required for both directions of media understanding.

Chapter 4

Description of Audio and Biosignals

Introduces the fundamental properties of audio and of biosignals, lists typical applications and media understanding solutions, discusses feature transformations applied on audio and shows that the same transformations can also be applied on biosignals.

4.1 Introduction and Dimensions of Hearing

The first three chapters were dedicated to the setting of media understanding. We have investigated the scope, the media types and the general structure of media understanding applications. In this and the next two chapters we investigate methods for the description of media. The focus is on simple methods that explain the fundamental concept of feature transformation. Still, most of the presented methods are used practically and have proven very successful in numerous applications.

The present chapter deals with the description of audio and of biosignals. Why do we start with audio? One reason is that audio is a structurally simple data type. A mono recording may capture music as well as speech and consists only of one stream of amplitude values. A second reason is the importance of the aural sense for the human being. Until recently, when due to the rising importance of visual media the visual sense became dominant, human culture was an oral culture, i.e. dominated by the sense of hearing. Even today, many non-western cultures are dominated by this sense. The visitor of sub-Saharan

Africa and of parts of Oceania will find that the radio and the mobile phone are ubiquitous in those regions while television is significantly less important. A third reason for starting with audio is the incredibly efficient implementation of the human sense of hearing. Outer and middle ear act as frequency amplifiers and band filters, the cochlea organ implements a transformation similar to the Fourier transform, and the brain can implement the sense of hearing by only 3.10^4 nerve cells – in comparison to 2.10^6 for vision and 4.10^7 for the olfactory sense (see Chapter 23 for more). Eventually, from the pedagogic point of view, starting with audio makes sense, because the methods used on other data types are almost all related to what is employed on audio. In particular, biosignal feature transformation is to a large degree a subset of what is applied in the audio domain. It makes, therefore, sense to include biosignal description in this chapter.

Though technically just a one-dimensional signal, audio has a variety of semantic dimensions. The most fundamental distinction is by the *type of content* into the following three groups:

- Speech
- Music
- Environmental sounds

The first two categories do not require explanation. The third, on the other hand, is a very heterogeneous group. It includes the sounds created by animals as well as traffic sounds. Further items are sounds created by machines, sounds created by humans that are neither speech nor music (e.g. laughter), literally environmental sounds (e.g. the booming sea) and many more. In the last section of Chapter 2 we have already seen that different types of content lead to significantly different technical characteristics. We have, for example, seen that speech signals are generally simpler, sinusoid signals than music signals. Environmental sounds are sometimes similar to music (e.g. whale sounds), but most times highly different from both speech and music and rather similar to EEG biosignals (e.g. β waves).

Next to the type of content, there are several other dimensions of audio. The following list of properties is certainly not exhaustive.

- *Tempo*: In speech, different subjects and environmental conditions require different tempo. The tempo of music is related to the type of music.
- *Rhythm*: Music is generally distinguished by the rhythm patterns that are the foundation of a particular melody. However, a good speaker will also try to induce a rhythm that increases the level of attention of the listener.

- *Melody*: Captures the arrangement of tones (pitches) based on the rhythm. In speech, bound to the phoneme structure of the language.
- *Harmony*: Describes the perception of simultaneous melodies, i.e. pitches. This property is highly relevant for music, but of course not for speech.
- *Timbre*: The overtone structure created by a sound-creating device (e.g. human voice, brass instruments).
- *Instrumentation*: The perception of simultaneous timbres. Like harmony, this feature is relevant for the understanding of music and, sometimes, of environmental sounds.

Each of these properties is independent of the others, i.e. one dimension. Though the tempo is an important factor for the perception of rhythm, the same rhythm can be combined with different tempos. The same melody can be played with different instruments creating different perceptions of harmony, timbre and instrumentation.

Next to these technical dimensions of audio, we also have to consider inner-human dimensions like the following.

- *Perception of complexity*: In particular, the perception of complex music (complex rhythm, melody, harmony, timbre, instrumentation) requires an expert listener. Sophisticated music may be perceived as unpleasant by the untrained ear while the expert may regard it as highest perfection.
- *Priming of the listener*: Different cultures have developed different models of sound, in particular, music. For example, the application of the Pythagorean tuning laws constitutes a division line between so-called western music and other musical styles. The priming of the listener is of highest importance for audio perception.
- *Mood of the listener*: The perception of audio fluctuates with the state of the human nervous system. For example, a tempo that is in one situation perceived as pleasant may be perceived unpleasant in another.

Obviously, the subjective inner-human dimensions are out of scope of audio understanding. These perceptual properties cannot be extracted from the media objects. Properties that can be captured include the following.

- Loudness
- Duration
- Pitch

- Rhythm
- Timbre

The first property is based on the *pressure level of sound*, frequently measured in *Bel*. Below, we will see that the human perception of loudness is not linearly equivalent to sound pressure. However, this property is important, for example, for distinguishing different types of environmental sounds. *Duration* is related to loudness. This property captures the amount of time that sounds of comparable loudness are perceivable.

Pitch and *rhythm* are also related. In the context of this book, we understand as pitch the recurring fundamental frequency of a short window of time. In contrast, rhythm shall be a time-limited sequence of amplitude peaks that reappears regularly over long windows of time. Eventually, the understanding of *timbre* here is, by and large, equivalent to the structure of overtones.

Comparing the list of describable properties with the dimensions of sound given above, we can see that the actually extractable properties are much simpler than the properties perceived by human beings. This fact is a typical example for the *semantic gap* in the feature transformation step of media understanding. Only a small amount of the semantic characteristics important to human beings can be transferred to the machine. The rest is ignored. It is, therefore, no surprise that many complex audio understanding problems have not yet been solved satisfactorily.

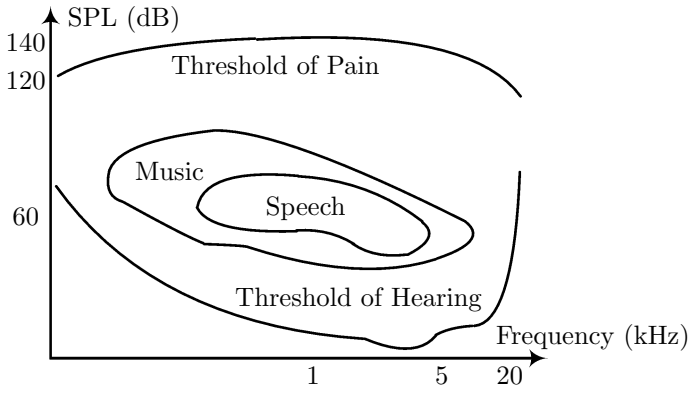


Figure 4.1: Thresholds of Hearing.

Before we move on to the actual feature transformations we consider it beneficial to introduce three fundamental bits of knowledge of psychoacoustics (see also Chapter 23). Figure 4.1 illustrates the thresholds of human hearing. These thresholds depend on the *sound pressure level* (SPL) and the *frequency* of the

sound. We can see that the best hearing is achieved (10dB and less) at frequencies around 4kHz. Please note that the frequency dimension is scaled logarithmical. Human hearing has a limit at around 20kHz. The threshold of pain lies somewhere between 110dB and 140dB.[1]

The two kidney-shaped objects in the center of Figure 4.1 represent the areas where speech and music can be perceived. We can see that speech uses significantly smaller bands of frequencies and amplitudes, which may be one reason why the speech understanding problem has been solved to a much more satisfactory degree than the problems of music understanding. Another reason, of course, is the relatively small number of sounds relevant in speech processing.

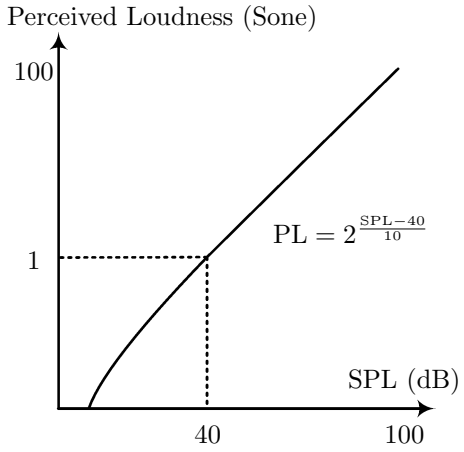


Figure 4.2: Perceived Loudness in Sone.

Figures 4.2 and 4.3 express two fundamental characteristics of human audio perception. The first graph relates the physical level of sound pressure that stimulates the ear-drum to the perceived loudness measured in *Sone*. Please note that the sone scale is logarithmic. We can see that the perceived loudness increases over-linearly, i.e. small increases in SPL cause large increases in perceived loudness. Below a certain threshold loudness is not perceived anymore.

The second graph relates the frequency of a sound to the perceived pitch measured in Mel. Below a frequency of 1kHz, the pitch increases over-linearly, i.e. doubling the frequency causes more than doubled pitch. Above 1kHz, the effect is reversed. Doubling frequency causes only under-linear increases of perceived

[1] As every parent will find, an unpleasantly low value. In the night before writing this paragraph the author could, for example, measure that his younger daughter could cry at 125dB. A higher threshold of pain would be desirable – but certainly cause adaptation in the offspring.

pitch.

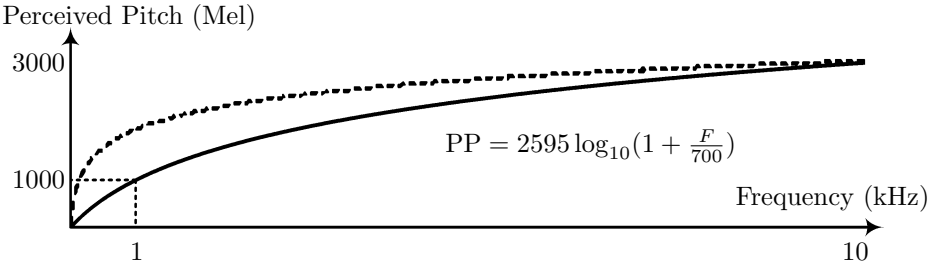


Figure 4.3: Perceived Pitch in Mel and Logarithmic Curve (dotted).

The sone transform and the mel transform are two fundamental truths of human sound perception. It is, therefore, important to consider them in feature extraction from audio. The given audio signal expresses amplitudes over time. These amplitudes need to be transformed into sone in order to represent the human perception correctly. From the amplitudes over time frequencies can relatively easily be computed. These frequencies need to be weighted by the mel transform in order to represent the human pitch perception correctly.

The remainder of this chapter is organized in three sections. In the next section, we introduce fundamental audio transformations as they are used in media understanding today. The subsequent chapter is dedicated to more advanced audio transformations that make use of the convolution operators discussed in the previous chapters. The last chapter introduces the transformations typically employed on biosignals and discusses the similarities between audio transformations and biosignal transformations.

4.2 Fundamental Audio Transformations

Below, we describe fundamental features for the description of loudness, duration and pitch. Rhythm and timbre descriptions are discussed in the next section. Please note that all the feature transformations we are going to describe are *recipes*, i.e. programs assembled from building blocks that have proven successful in practical application. The design of feature transformations is not a scientific undertaking but an engineering process. There is, therefore, no point in trying to prove one specific feature transformation. We will rather argue why the chosen building blocks should be useful in order to solve a particular problem.

From a rigorous point of view, all audio descriptions are local descriptions and do not belong to this chapter but to Chapter 14. This is because almost all audio descriptions are not extracted from the entire audio signal but from

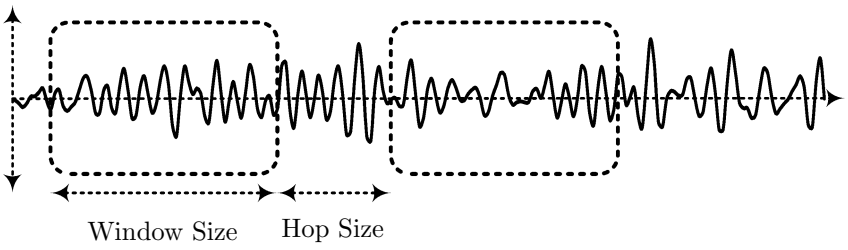


Figure 4.4: Localization by Windowing.

time-limited *windows*. Figure 4.4 illustrates two windows. The windows consist of the samples inside the time limits *begin* and *end*. Windows are typically defined by two parameters. The *window size* defines the number of samples considered by the feature transformation. Typical window sizes are 20ms (e.g. about one phoneme of speech), 40ms (equivalent to one frame of 25 frames-per-second video), 1s (chunks of music, environmental sounds). The *hop size* defines the number of samples not considered between two windows. The hop size may be zero (all samples are considered), smaller than zero (some samples are considered in more than one description) or larger than zero (some samples are ignored).

In audio feature transformation, the window size is often fixed and, in consequence, the overall size of the description would vary with the size of the media object. This undesired property can be avoided by normalization of the media objects under consideration prior to feature extraction or by defining a maximum length description and filling up empty spaces of shorter media objects with zero values.

Before jumping into the pool of features, we should reflect the properties desired from a description which is used as input to categorization. If a set of media objects should be categorized into a finite set of classes, the feature transformation step should advance the process as far as possible, i.e. create descriptions that are as similar to the class labels as possible. For this end, it is desired that media objects belonging to the same class have highly similar descriptions. That is, the variance of corresponding description elements should approach zero. Additionally, descriptions of media objects belonging to distinct classes should be as different as possible. That is, the variance of corresponding *description elements* should be maximal. These are the standards by which feature transformations should be measured in media understanding. See Chapter 11 for more.

For a start, we investigate three examples for loudness features.

- Short-time energy

- Other simple statistical features
- Logarithmic attack time

In all three cases, we want to measure the energy of the signal. Such a feature transformation may be valuable for the categorization of pieces of music into genres (e.g. pop music vs. heavy metal), the recognition of emergency situation (raised voice, alarm signals), etc. One of the most frequently used feature transformations is the *short-time energy* which is defined as follows ($o \in O = [s_l | s \in S \wedge l \in L_{time}]$ as defined in Chapter 2):

$$f_x = \frac{\sum_{i=0}^{h-1} s_{i+x*h}^2}{h} \tag{4.1}$$

The description $f \in F$ holds the squared (audio samples have a sign!) sum of all samples in window x of size h. Please observe that this formula does not consider a hop size. Alternatively, the following short-time energy is used as well:

$$f_x = \frac{\sum_{i=0}^{h-1} |s_{i+x*h}|}{h} \tag{4.2}$$

Though the values and their relationships are different, the alternative definition practically leads to the same result. The embedding of the above equations in a feature transformation algorithm could be performed as follows:

```
h:=16000/40
for x:=0:size(o)/h
   f(x):=0
   for i:=0:h-1
      f(x):=f(x)+|s(i+x*h)|
   endfor
endfor
```

This simple algorithm makes use of the second equation. It describes an audio object o sampled at 16kHz in chunks of window size 40ms, i.e. 400 samples. The result is stored in the description vector f. This algorithm provides a prototype for most feature transformations discussed below since it *summarizes* the input signal. We will, therefore, refer to it where necessary.

Alternatively to short time energy, a number of statistical moments could be employed for capturing loudness. Suggestions include the median of the squared

samples (less prone to outliers than the mean), the maximum amplitude, the average of the n largest amplitudes, the number of samples above the mean, etc. These moments could be weighted by a *belief score*, for example, the span of samples, variance or standard deviation – of the entire window or just the samples above the mean, the top n samples, etc. Statistical moments are used in many feature transformations. The decision, which moment to use has to be made empirically. If one particular moment performs well on media objects characteristic of the media type under consideration, it is advisable to consider it for feature transformation.

Psychoacoustic remark. In the above definition of loudness, we did not consider the psychoacoustic facts depicted in Figure 4.2. For practical usage of moments of loudness it should be beneficial to apply the sone transformation on the samples before summing them up.

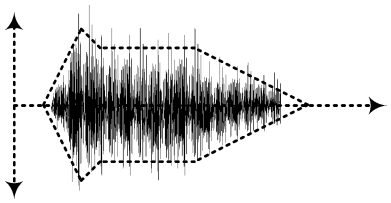


Figure 4.5: The Attack-Decay-Sustain-Release Model (ADSR) of Sound.

The third loudness feature that we want to discuss was defined in the MPEG-7 standard for media description [191]. It assumes a particular model of sound (in terms of amplitude) depicted in Figure 4.5. According to the ADSR model each sound consists of four phases: the attack phase where the amplitude increases strongly, the decay phase where the amplitude decreases a little, the sustain phase that holds the amplitude and the release phase where the sound fades out. The *log attack time* feature transformation uses this model for identifying attack phases of sounds in audio signals. In order to measure the attack time it is sufficient to detect amplitude peaks, i.e. samples with amplitudes significantly higher than all other samples in a to be defined neighborhood. From a detected peak, the log attack time can be approximated by searching backward in time to the next minimum, extrapolating the position of the zero crossing of this sound and taking the logarithm of the time span between extrapolated start and amplitude peak.

Criticism. Log attack time is not necessarily a loudness feature. It describes the time that a sound needs for reaching its maximum but does not express the magnitude of this maximum. However, empirically log attack time correlates negatively with loudness. Quickly attacking sounds are perceived as sharper, louder sounds. That is, from a psychoacoustic point of view log attack time

does indeed measure perceived loudness. Taking the logarithm helps this process by stressing differences in magnitude in attack time. Small differences are reduced. Large differences are emphasized. Additionally, nothing can be said against adding the magnitude of the attack peak to the description – as some kind of belief score for the attack time (higher peak=higher belief in the importance of this particular attack time). Eventually, the reader has to be aware that the extrapolation of log attack time is non-trivial since most audio media consist of overlapping sounds. The interesting sound (signal) will, therefore, be hidden (masked) under many irrelevant sounds (noise). That is, one fundamental problem of log attack time computation is a generally bad *signal to noise ratio* (see Section 4.4).

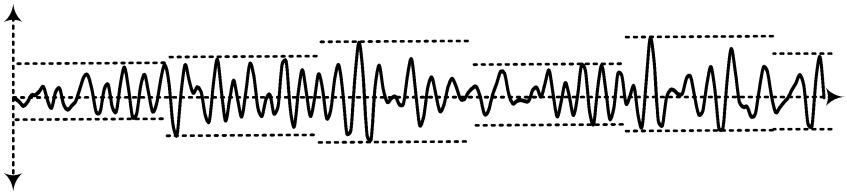


Figure 4.6: Description of Duration.

Before we turn to the description of pitch, we would like to introduce a descriptor for the duration of segments of similar amplitude. This descriptor has turned out valuable for the categorization of various kinds of environmental sounds. Figure 4.6 illustrates one possible implementation. The dotted lines stand for the description values. Segments are described by maximum amplitude at fixed length intervals and by the span of maximum and minimum amplitude. Another – practically superior – formulation would be to start from one particular amplitude, extend the duration until maximum/minimum of amplitude exceeds a pre-defined *threshold* (e.g. 10%) and use the duration value as description. The advantages of this approach are higher accuracy and flexibility concerning the characteristics of the media type. The disadvantages are the need to set a parameter, and the possibility that the feature transformation process ends up with a long non-discriminative list of short duration values that form an irregular description vector.

Please note that at this point we touch a fundamental problem of feature transformation already for the second time. Starting from the semantic description of an interesting media property (loudness, duration), a scheme for feature transformation can be developed that should precisely represent it. Such accurate models, however, regularly require parameterization and, in practice, surprisingly often turn out inferior to less accurate models that cannot be linked to the semantic media understanding goal by straightforward reasoning. The link is

mostly only statistical. Nevertheless, in media understanding such a statistical link is always preferable if it results in better descriptions.

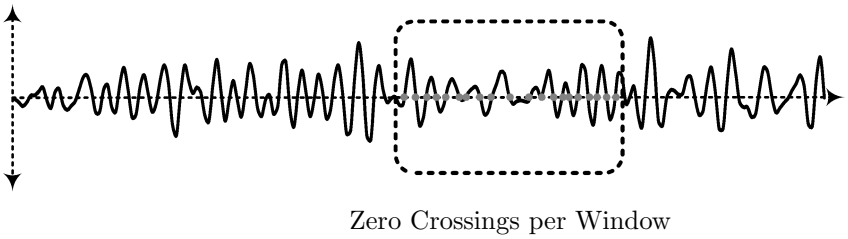


Figure 4.7: Zero Crossings Rate.

Figure 4.7 illustrates a simple yet powerful feature transformation for pitch description: the *zero crossings rate* (ZCR). The ZCR is a measure for the fundamental frequency of a window of sound. It simply counts the number of pairs of samples with alternating sign. If this sum is large, we assume a high fundamental frequency, if it is low, we assume a low fundamental frequency. From the fundamental frequency of a short window of sound we have a direct link to the pitch. The higher the fundamental frequency, the higher the pitch. In order to meet with psychoacoustic insights it is, furthermore, advisable to apply the mel transformation on the ZCR values, i.e. to transform the fundamental frequency into perceived pitch.

The ZCR – though simple and fast to compute – is of highest significance for many audio understanding applications. Often, it has proven to hold information that cannot be extracted by complex spectrum-based descriptions as the ones discussed in Chapter 13. In conclusion, it is advisable to include the ZCR in any description applied on audible media even if a reasonable semantic explanation cannot be given.

As a second feature transformation for the recognition of the fundamental frequency we would like to mention the *peak histogram*. This feature employs autocorrelation in order to approximate the pitch of sounds. Since autocorrelation is based on convolution this feature transformation will be explained in the next section.

We would like to close this section with two remarks. Firstly, the presented feature transformations rate among the simplest used in audio description. They were chosen because of their simplicity as introductory examples but as well, because they are really used in state-of-the-art audio understanding applications. In particular, short time energy and zero crossings rate are important features for speech, music and environmental sound categorization. More audio feature transformations can be found in the excellent source [268].

Secondly, we would like to touch the question of multiple channels. Audio, in particular, is often available in stereo or better quality, i.e. two or more channels exist. What can we do with those channels? Generally, three approaches are thinkable:

1. Throw away all channels but one. An example scenario could be speech recognition from a good-quality source.

2. Compute descriptions for each channel and use all descriptions. This strategy makes sense if the channels contain different information (e.g. music recordings with the vocals on one channel and the instrumentation on the other).

3. Compute descriptions for each channel and merge them statistically. For example, the mean could be used for description and the standard deviation as a belief score. This strategy could turn out beneficial in scenarios where different channels are exposed to different sources of noise (e.g. in a traffic surveillance system).

Which approach to take depends on the characteristics of the involved media. Please refer to Chapter 7 for a deeper discussion of this problem.

4.3 Audio Description by Convolution

So far, we have investigated feature transformations that employ statistical operations on *fixed-size* windows of samples in order to build up a description. This approach works well for the description of loudness, duration and fundamental frequency. Complex semantic categories such as rhythm and timbre, however, cannot be described properly by statistical moments. Therefore, the transformations discussed in this section implement a fundamentally different idea. They employ the convolution operators on windows of *variable* size in order to identify extremes that express the desired media property.

An example should make the idea clear. Figure 4.8 depicts two waves: Signal (a) is a low-frequency sound of sinusoid shape (no overtones). Signal (b) is also a pure tone, but of higher frequency. The two windows encapsulate exactly one sine wavelet each. As can be seen from the figure the window of signal (a) is wider than the window of signal (b), precisely $h_a = 2h_b$. That is, the window size h is a proper description for the media pitch, since the larger the window size the lower the pitch is.

The only problem of this approach is identifying the window size that encapsulates exactly one basic wavelet. Real-world audio signals are based on complex instrumentations, including sources with complicated overtone structures as well

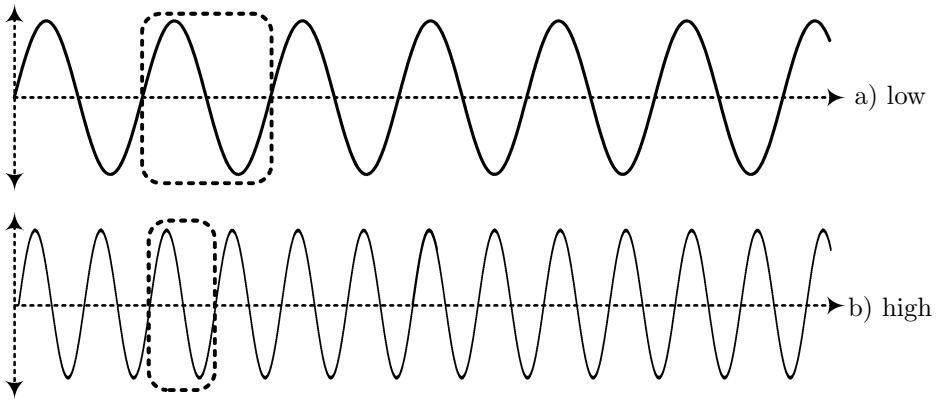


Figure 4.8: Autocorrelation and Window Size.

as noise. Identifying the right h is not as easy then as for pure tones. The approach commonly taken is based on convolution. The feature transformation *pitch histogram*, for example, implements the following model where windows $o_i = cut(o, ih, (i+1)h-1)$ of size h are cut from an audio object $o \in O$ with locations set L.

$$f_x = \arg\max_{h \to 1} \sum_{i=0}^{\frac{max(L)}{h}-1} o_i \otimes o_{i+1} \tag{4.3}$$

The implementing algorithm starts with a small window size and reduces h in the process towards zero. For each window size the convolution of the resulting windows is computed using the inner product and eventually summed up. Of the resulting sums the one with the highest value h_{max} is chosen and used as description. Why the highest value? The inner product is maximal for two identical input objects. Hence, the maximum value indicates that windows of this size encapsulate the most basic reappearing object, i.e. the sought wavelet.

This approach is commonly referred to as *autocorrelation* since it applies the means of correlation (here, positive convolution) on fractions of the same signal. The pitch histogram is a feature transformation that identifies the fundamental frequency of a signal like the ZCR does. However, since ZCR can be computed much faster, the pitch histogram is not as frequently used as the ZCR.

One paramount feature for the measurement of rhythm in audio signals is *linear predictive coding* (LPC). In fact, LPC as a method is not just applied in the audio domain but as well on other data types such as stocks, biosignals and various forms of statistical data. There, it is usually used for the prediction of future values based on a given data set – which is the origin of the name.

On audio it is typically applied as follows (same objects given as for the pitch histogram):

$$f_x = \arg\min_{h \in \{x:y\}} \sum_{i=0}^{\frac{max(L)}{h}-1} o_i \bar{\otimes} o_{i+1} \tag{4.4}$$

The formula appears highly similar to Equation 4.3. In fact, the ideas of LPC and pitch histogram are the same. However, the implementation is different. LPC uses negative convolution instead of positive convolution. That is, the operator is most sensitive to differences instead of similarities. The negative convolution becomes minimal for identical objects. Hence, the utilization of the minimum instead of the maximum. In Section 3.3 we already introduced negative convolution based on the L_1 metric. For LPC, the operator is usually based on Euclidean distance:

$$x \bar{\otimes} y = \sqrt{\frac{\sum_{i=0}^{n} (x_i - y_i)^2}{n}} \tag{4.5}$$

Since this formulation is equivalent to the least squares method in regression the LPC is also referred to as *auto-regression*. What is the difference between autocorrelation by convolution based on the inner product and auto-regression by convolution based on the Euclidean distance? According to psychological research (see Chapter 28 for details) one major difference is, that the inner product models how humans perceive similarity in simple structures (so-called separable stimuli) while the Euclidean distance models the way we perceive the difference of complex structures (so-called integral stimuli). Hence, the latter method should be superior on structures more sophisticated than wavelets.

Indeed, the second difference between LPC and pitch histogram is that the latter does not start window size h at a comparably small value and moves it towards one – rather, h iterates linearly through a set of values between the relatively high limits x, y. In consequence, LPC is sensible to recurring windows of comparably high complexity. If the limits x, y are chosen properly 'recurring windows of comparably high complexity' is a fair definition of rhythm. That is why LPC is employed as a feature transformation for rhythm detection.

The last feature transformation based on convolution that we would like to describe here is the *audio harmonicity* transform defined in the MPEG-7 standard. Audio harmonicity implements the following idea:

$$f_x = \sigma\left(\left\{ \sum_{i=0}^{\frac{max(L)}{h}-1} o_i \otimes o_{i+1} \middle| h \in \{x:y\} \right\}\right) \tag{4.6}$$

That is, for a given range $x : y$ of window sizes the inner products are computed and the standard deviation of the averages for all considered values of h is employed as description. Why this algorithm? Timbre is the structure of overtones particular for a certain sound-generating source (e.g. instrument). Audio harmonicity captures this property by computing the self-similarity of the signal at various levels h and summarizing these self-similarities by their variance. Alternatively, another statistical measure of variation or even the entire set of values could be employed as description. This timbre feature is based on the idea of the correlogram, a method that will be described in the next section since it is frequently used on biosignals.

The presented feature transformations are just examples. LPC and pitch histogram are frequently-used methods while the timbre is often captured by spectral features (see Chapter 13). However, all of them are excellent examples for the modeling of signal self-similarity by convolution. Summarization and convolution are – as we will see – the two principal approaches in feature transformation. These feature transformations together with the statistical feature transformations introduced in the last sections are able to provide powerful descriptions for all sorts of audio understanding applications. In the next section we will see, how related methods are employed for the description of biosignals.

4.4 Biosignal Feature Transformations

This section is organized as follows. First we investigate the technical process of biosignal capturing. Then, we outline important applications based on biosignals. Eventually, we introduce feature transformations suitable for these applications and discuss their similarities with audio feature transformations.

The machine understanding of audio and biosignals are related domains even though the applications do not express this relationship. From Figure 2.2 we can see that the methods employed for biosignal understanding are a subset of what is employed on audio. As a matter of fact, most audio feature transformations are similarly applied on biosignals. The major difference between the domains is in categorization where more sophisticated methods are used on audio today. Technically, audio and biosignals are isomorph. The dimensions and types of samples are the same, the sampling rates highly similar. Typical bandwidth requirements of biosignals lie between 1kHz (ECG) and 10kHz (EEG).

Biosignals are usually measured by electrodes that are sensitive to changes of potentials. Since this approach leaves the surface of the body intact, it is called non-inversive. Inversive methods such as electrocorticogram and intracorticogram shall not be discussed here, because they are hardly ever used for media understanding (rather, for clinical purposes). The typical non-inversive setup comprises one to many electrodes for the signals of interest that are measured

against a baseline (bi-polar capturing). For example, for ECG (heart signal) capturing two electrodes are sufficient. For EEG (brain signals) capturing 20 and more signal channels are typically used. Other signal types include EOG (eyes) and EMG (muscles) with similar numbers of channels. Before feature transformation the potentials are usually amplified externally (passive electrodes) or in the (active) electrodes. Typical bandwidths of the signals range from 0.2Hz (respiration) over 1Hz (EEG) up to 5Hz (heart rate).

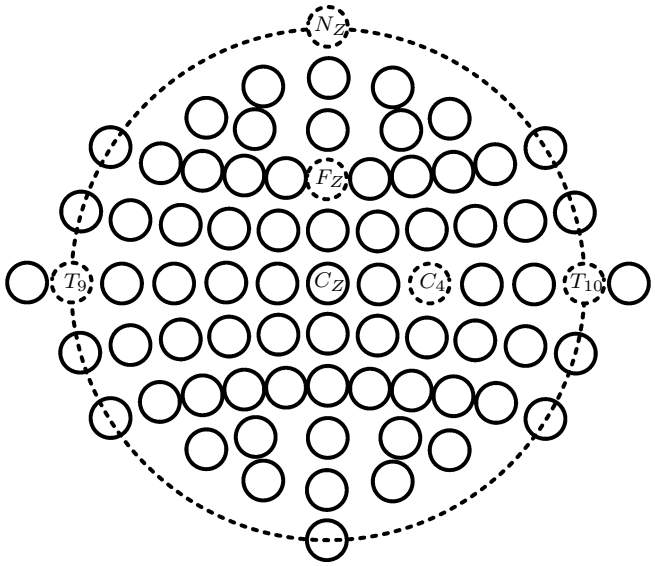


Figure 4.9: The 10-20 EEG Capturing System.

For the capturing of brain signals for scientific purposes, an international standard for the layout of the electrodes exists: the 10-20 system. It is depicted in Figure 4.9. The name is derived from the angles between geodesic lines on caps implementing this convex scheme. For example, the angle between electrodes F_Z and C_Z is 20 degrees. The 10-20 cap is positioned on the human head with point N_Z above the nose and points T_9, T_{10} above the ears. The standardized naming of electrode positions facilitates the exchange of research results. Similar systems exist for EMG capturing.

Different types of biosignals have different properties. See Figure 4.10 for examples. The EEG α wave is quasi-periodic and harmonic (smooth). The β wave (taken from point C_4) is neither harmonic nor periodic. The ECG signal, on the other hand, is not harmonic but quasi-periodic. It is furthermore, a pulse signal (periodic spikes that express the contraction of the heart muscle do exist) and has a baseline to which the signal returns after each pulse. These properties

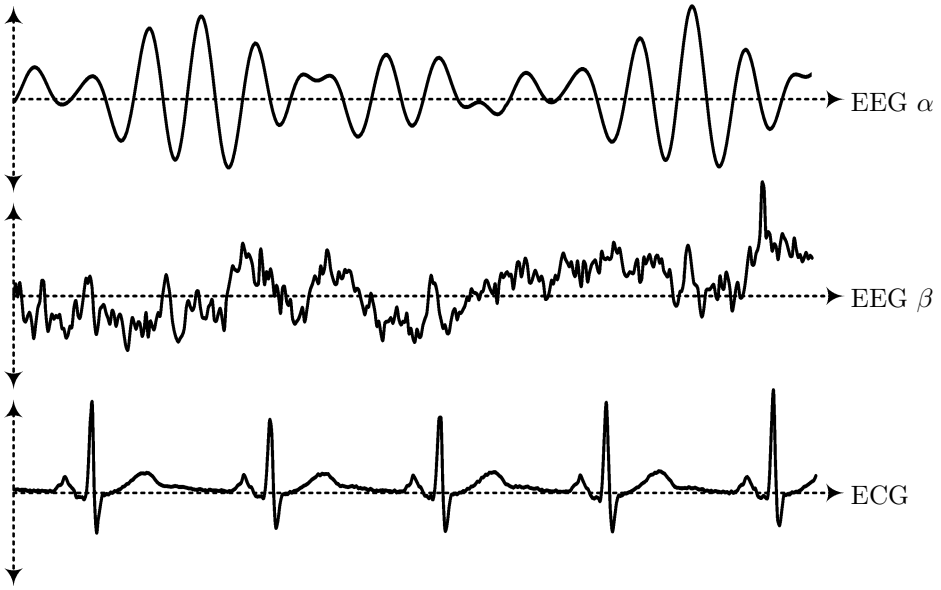


Figure 4.10: Types of Biosginals.

are essential for the understanding of the signal. Different characteristics require different feature transformations and allow for the implementation of different applications. It makes, for example, no use measuring the energy of an ECG signal. However, the fundamental frequency of the heart signal could be captured by the ZCR.

One particular property of EEG signals is their extremely bad signal to noise ratio. An average electrode has a receptive field of $7mm^2$. Considering the neuronal density of the brain and the size of its surface, a single electrode may capture the potentials of up to 5.10^4 neurons! Since only a small number of neurons is usually employed in one particular cerebral process (signal) the vast majority of the potentials merged in one electrode are noise. Under these circumstances only very general applications can be implemented.

Most biosignal processing applications are based on EEG waves. Non-EEG applications include the measurement of excitement (for example, by heart rate, skin resistance sensors) and the measurement of exhaustion (e.g. pulse sensor, ECG). These applications are only of minor interest for media understanding. Interesting and relevant EEG applications include the recognition of:

- *Steady-state visual evoked potentials*: These are significant signal peaks

caused by the presentation of unexpected or desired visual stimuli. A typical application would be the presentation of images of food to a locked-in person and selection of the one dish that causes the highest potential. Since such peaks are observed typically 300ms after presentation of the stimulus the problem is also called *P300 recognition*.

- *K complexes*: K complexes are positive potential peaks followed by equally-sized negative potential peaks. They are very similar to evoked potentials and typical for non-REM sleep. The application is therefore, categorization of sleep phases.

- *Slow cortical potentials*: These are amplitude changes of very low frequency. Applications lie in the clinical analysis where the correlation between changes in the blood flow and slow cortical potentials could be shown.

- *Changes of oscillatory activity*: Every neuron oscillates. Changes in oscillatory activity become visible as changes in frequency and amplitude. They express a change in synchronization and desynchronization of neural complexes and are, for example, employed for the recognition of the symptoms of epilepsy.

- *Real/virtual motor activity*: Motor activity becomes visible as particular patterns over several EEG channels in form of amplitude and frequency changes. The recognition of virtual motor activity can, for example, be employed for the movement of protheses and wheelchairs.

Most of these applications are targeted at clinical purposes. The most interesting one for media understanding are certainly P300 recognition and motor activity recognition. For the recognition of such patterns, we have the following groups of feature transformations available:

- Amplitude features

- Spectral features

- Template features

The two latter groups are discussed in Chapters 13 and 24, respectively. The first group are those feature transformations that generate descriptions directly from the potentials captured by the electrodes. Naturally, these features are very simple. They allow for the recognition of the same types of events as their relatives employed on audio: *energy*, *fundamental frequency* and *peaks detection/rhythm*, in particular. Energy can be employed to identify motor activity,

fundamental frequency for the recognition of slow cortical potentials and oscillatory activity and peak detection/rhythm for the recognition of evoked potentials and K complexes. In the next paragraphs, we will investigate some feature transformations for these purposes.

Generally, biosignal feature transformations employ windowing. Typical window sizes h are 200ms (ECG), 1s (EEG) and 5s (respiration). Obviously, the window size correlates with the bandwidth and is considerably larger than what is used in audio understanding. Audio signals are usually denser, i.e. of higher frequency. However, the methods applied on these windows are highly similar to audio understanding. The energy of a window of potentials is computed by the short time energy algorithm (used for loudness on audio). Alternatively, statistical moments are employed for energy description. The fundamental frequency is usually extracted using the zero crossings rate, only that the method is sometimes called *period counting*.

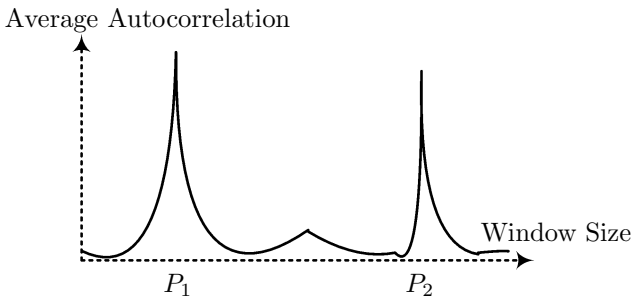


Figure 4.11: A Typical Correlogram of a Biosignal.

The problem of peak detection and rhythm detection is of significantly higher relevance for biosignals than the description of energy and fundamental frequency. It is commonly referred to as *periodicity recognition*. Two fundamental approaches are the *Pan Tomkins* feature transformation and the *correlogram*. The first method recognizes signal peaks by the following algorithm:

1. Compute the first derivatives of the samples. That is, each pair of neighboring samples (s_i, s_{i+1}) is represented by $s_{i+1} - s_i$.

2. Remove the sign by squaring the derived signal.

3. Select values close to zero as peaks (low-pass filter).

Frequently, the Pan Tomkins approach employs a Barkhausen band filter for pre-processing (see Chapter 23). The first derivative of an extreme (discrete) signal will be (close to) zero. The same idea for peak detection is – as we will

see in Chapter 14 – often employed on visual media. The peaks detected by the Pan Tomkins approach can be used to describe the periodicity as, for example, mean and standard deviation of the intervals between pairs of peaks. The Pan Tomkins approach works well on smooth signals.

An alternative, visual method popular in biosignal processing is the *correlogram*. Figure 4.11 illustrates an example. The correlogram is usually constructed by autocorrelation based on the inner product. Like in the case of the pitch histogram the window size is varied from a reasonable starting point close to zero. At every window size, the average autocorrelation of the signal is computed and printed in the correlogram. That is, a peak in the correlogram indicates a recurring rhythm pattern at this window length. A correlogram without peaks indicates a random signal. Implementing the correlogram as a description is equivalent to the pitch histogram. The correlogram method works well on quasi-periodic and pulsed signals.

Interestingly, linear predictive coding is hardly used in biosignal understanding. This is surprising since statistical methods, including regression are of highest significance in biosignal processing. For the scientist, it may be interesting to base P300 recognition-based media understanding on linear predictive coding, since this method performs so well in the audio domain.

In summary, biosignal understanding can be implemented with feature transformations very similar to those applied on audio. The usage of non-inversive methods opens some very interesting perspective for multimedia understanding. The recent development of affordable hardware for this purpose should result in widespread application of such applications in the near future.

Chapter 5

Description of Visual Media

Discusses fundamental properties of visual perception, including the representation of colors in color spaces and the extraction of edges by edge operators, introduces feature transformations for the description of color information, textures and shape information.

5.1 Properties of Visual Perception

This chapter is about a classic of media understanding, known as content-based image retrieval. The techniques introduced here have been developed over three decades, and since numerous approaches were suggested over the years we can, of course, only present a selection of outstandingly successful methods. Everything discussed in the next sections is applicable to both images and videos. The methods fall into three groups that name the three remaining sections: description of color, texture (surface properties) and shape information. Before we discuss algorithms in the next three sections, we would like to introduce the major properties of the human sense of vision in this section. We selected a few important physiological properties and how they are modeled in media understanding. More information on the sense of vision can be found in Chapter 23.

Before we start with a systematic description we would like to give an example. Figure 5.1 illustrates a holistic visual feature named *visual keywords*. The description is simply made up from rectangular subregions of the media object. Every rectangle is a visual keyword. The set of visual keywords represents the media object. Visual keywords carry color information on the pixel level, tex-

Figure 5.1: Description by Visual Keywords (© *CNBC*).

ture information (i.e. information on the common properties of pixel groups) and local shape information. The latter type of information, however, is usually poorly represented by visual keywords due to *random selection* of rectangular regions of the media object. The visual keywords approach was popular in the late 1990ies. Its major strength is its simplicity. The major weakness is a tendency to false positive matches. The figure gives examples. The arrows indicate matches of media object and visual keywords. The dotted arrows indicate incorrect matches. Such false positives can easily occur because the information in visual keywords is often as redundant as in the media object. Still, the approach is regaining popularity – this time coming from the text domain where it is known as the *bag of words* method. In visual information retrieval, it is now called *bag of features* approach. See Chapter 14 for details.

Figure 5.2: Saccadic Scanning of Visual Content (© *CNBC*).

We have started with this example, because it provides the floor for a discussion of some fundamental properties of human visual perception. Large proportions of the human brain are dedicated to the processing of visual information. The type of information that matters most for human vision is contrast. Research could show that the first stages of visual processing are concerned with the detection and categorization of edges (i.e. contrast). This property distinguishes the sense of vision from the sense of hearing where harmonic patterns are of the greatest interest. Following this insight the paramount success factor

of visual keywords is to capture unique patterns of contrast in the media object while neglecting its uniform parts. In this sense, the visual keywords in Figure 5.1 are well chosen.

A second important property of human vision is the way the input signal is generated. The human eye is not able to capture a view like a camera that takes a shot at one moment of time or like the compound eye of a fly that takes multiple parallel images. It rather scans its environment and produces a steady flow of information. This phenomenon is called *saccadic seeing*. Figure 5.2 shows an example. The black lines indicate the movement of the focus of one eye over time. As can be seen, high-frequency information is scanned more often than areas of uniform pixel values. In particular, structures recognized by the sense of vision are scanned in detail (e.g. the eyes). In consequence, small regions of the image produce the highest proportion of the visual information stream. The visual keywords approach can imitate this property of human vision if it is able to capture the most relevant structures in the input object.

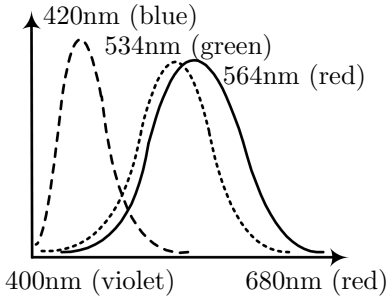


Figure 5.3: Cone Responses on Wavelengths of Light.

The next property of human vision that we would like to discuss here is color perception from wavelengths of light. The human eye has separate receptor cells for brightness (rods) and three different wavelengths of color (cones). The majority of these cells can be found on the fovea, the area around the optic nerve. Since the rods outnumber the cones significantly our sense of seeing has a much higher resolution for grayscale images. That is why ambitious photographers still prefer this 'old-fashioned' medium. The cones are sensitive to the wavelengths illustrated in Figure 5.3. The perception of blue is well-separated from the two others and maximal at a wavelength of 420 nanometers (nm). However, perception of red has its peak at 564nm – less than 6% from the green peak. This phenomenon is explained by the relatively late evolutionary diversion of the non-blue cone into a green-receptive and a red-receptive cone. Of course, during the majority of time of human evolution, the perception of green things (in particular, plants) was significantly more important than the perception of

red things. Today, the closeness of the green and red cones is responsible for some forms of color blindness. In summary, our sense of color has three input channels, namely the optical RGB (red, green, blue) nerve.

It is very pleasant for the computer scientist to see that the human eye uses the same *color model* as a computer. However, two remarks have to be made on this issue. Firstly, the highest sensitivity of the green and red cones is not placed exactly at the wavelengths of pure green and red. In fact, red perception takes place at significantly higher frequencies than 564nm. Secondly, the *three stimuli theory* says that any base of three dimensions would be able to provide a space big enough for the representation of all colors. The RGB model is just one color model. Mathematically equivalent color models are the HSV model, the YCrCb model and many others [348]. Please note that, though mathematically equivalent these models have properties that make one for a particular application more useful than the others. This practical usefulness, however, is only of limited interest for media understanding, since we want to extract information from the color information and this can be done from all color models with three dimensions.

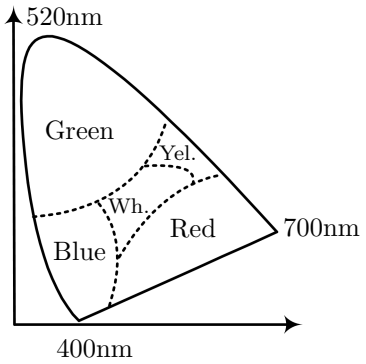


Figure 5.4: The CIE Color Space.

The three stimuli theory may appear surprising to the hobby physicist. Light has the properties of particles since it is emitted in photons and the properties of waves since the movement of these photons has a wavelength. Hence, it should be possible to define a two-dimensional color space that covers human perception fully. The CIE XY model – illustrated in Figure 5.4 – is such a color space. In two artificial dimensions (that is, they have no semantic meaning) it defines a color space with white in the center and the colors and color gradations in the periphery of the so-called color sail. The outside of the sail is black. The border of the sail corresponds to the wavelengths of the light ranging roughly from 400nm to 700nm. As can be seen, the largest area is covered by tones of

green. The success of the CIE model shows that the three stimuli theory defines a sufficient criterion for color spaces that is beyond necessity. That is, the other color models such as RGB contain redundant color gradations that cannot be differentiated by the human eye. For media understanding of visual material it is, therefore, highly advisable to convert the media objects into the CIE XY color model before the feature transformation step is executed! From RGB to XY the conversion goes as follows:

$$X = 0.431R + 0.342G + 0.178B \tag{5.1}$$

$$Y = 0.222R + 0.707G + 0.071B \tag{5.2}$$

One last aspect of human vision that we would like to mention here is related to the perception of texture and shape. In his interesting paper [126] Goldmeier states that the foremost strength of human vision is to distinguish the form (shape) and material (texture) of visual stimuli. The investigation uses quantitative analysis of the results of psychological experiments where students were shown figures with different shapes and textures and asked to rate their similarity. Our distinction of texture and shape can, therefore, be regarded as somewhat *natural*, since the human brain – a product of evolution – has come up with the same distinction.

We would like to close this section with a few notes on technical properties of visual media objects. It is certainly beneficial to keep them in mind when designing a media understanding application for visual media.

1. The lower resolution of video frames is not necessarily a disadvantage in visual feature transformation. Color descriptions and shape descriptions can be computed faster. Rather, a problem is the – often – reduced color depth provided by video cameras. The higher resolution of images is an advantage, however, for the description of textures. In consequence, it appears advisable to start color and shape extraction from images with a pre-processing steps that reduces the resolution.

2. Visual stimuli are *compositions* of objects. A major factor influencing the complexity of such compositions is whether or not the objects are rigid. If yes, more detailed representation may make sense than if not, because the higher variability of non-rigid objects introduces a source of noise that can be reduced in magnitude by coarse representation (i.e. reduced resolution).

3. Perspective is another influence factor on the composition. It is important for successful feature transformation to take into account under which perspectives the objects in the composition may become visible. Again, the more degrees of freedom exist, the more general the description should be in order to be useful for categorization.

4. The last two items on this list have already referred to the general problem of noise, distortions and missing data. Visual perception is exceptionally prone to occlusions. If parts of the composition are not visible, the information is simply lost for the description process. One remedy for this problem is localization of the descriptions. That is, to apply the feature transformation not on the entire media object, but on regions within the media object that are unlikely to be occluded. Visual keywords is one stochastic attempt to achieve this goal.

The art of media understanding is to avoid these problems where possible and to minimize their influence where they cannot be avoided. Below, we introduce feature transformations that have proven successful in this respect. The remainder of this chapter is organized as follows. The next section introduces feature transformations based on color information. Section 5.3 discusses the description of texture properties. The last section introduces general concepts for shape description.

5.2 Color Descriptions

One of the outstanding abilities of human beings is the perception of color. Color allows us to recognize and distinguish objects. Without color, some of our finest achievements, in particular, the visual arts would only be of little significance. It is therefore hardly surprising that the representation of color is of highest importance in visual media understanding. In this section, we introduce methods for the extraction of dominant colors, color distributions and local color properties. Most of these methods were already developed in the 1990ies – for media understanding very long ago, but since they are easy to compute and efficient to employ, they are still used today. Some of the methods were standardized in the MPEG-7 standard [243]. We give a brief description of these feature transformations and estimate the quality of their descriptions.

Before we start describing the feature transformations it is worth noting that – though the domain appears different – the methods used here are actually very similar to those used on audio and other data types. In the last chapter, we introduced the short-time energy feature transformation as a sequence of windowing, sample-wise elimination of the sign and summarization of these values. The same steps are usually taken in the visual domain. Windowing is some kind of *localization*, a method discussed in this section and, generally, in Chapter 14. The elimination of the sign is an example for *quantization*, i.e. the general transformation of the samples. The last step is an example for *aggregation*. That is, the samples are merged into a few values with statistical properties suitable for media descriptions. Below, we will see that the same steps are taken in the

visual domain. Please refer to Chapter 11 for a thorough investigation of the *building blocks* of media understanding.

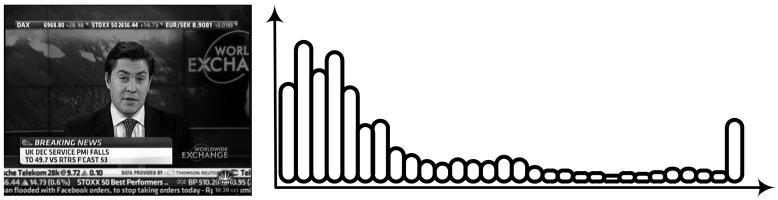


Figure 5.5: Example of a Luminance Histogram (© *CNBC*).

The most frequently used color descriptor is the *color histogram*. The idea is very simple. First, we define a set of colors (e.g. white, gray, black; red, yellow, white, green, blue, black). Then, each pixel of the visual media object is assigned to the color bin (e.g. white) with the highest similarity. Figure 5.5 gives an example where the pixels in the source image are aggregated in a color histogram of gray values. The resulting distribution is called a *histogram*. The general algorithm for the function introduced in Section 3.4 may be written as follows:

```
function color_hist takes x begin
   z=()
   foreach l in L(x) do
      y := quant_col(x(l), GRAY)
      z(y) := z(y)+1
   endfor
   return z
end
```

Here, x is a (fraction of a) media object. The locations l are drawn from the media object's location set $L(x)$. Observe that the loop implements a localization operation. The pixel colors are quantized into y and the histogram z aggregates the quantized data. The quantization function for the transformation of pixel values into distinct colors may be defined as follows:

```
function quant_col takes x, y begin
   if y = GRAY then
      z := 0.6 * chn(o,RED) + 0.3 * chn(o,GREEN) +
           0.1 * chn(o,BLUE)
   elsif y = RAINBOW then
      z := get_hue(x)
```

```
   elsif y = ...
      ...
   endif

   return round(z,8)
end
```

This function creates a color value rounded to 8 bits of accuracy. If the second parameter requests grayscale the output is a luminance value, and the computed color histogram is actually a luminance histogram. If the option *RAINBOW* is selected the resulting color histogram resembles the *scalable color* feature transformation of the MPEG-7 standard. This transformation extracts a color histogram of 256 bins from the *HSV* color space (hue, saturation, value). The final description is cleverly quantized and transformed by Haar wavelets (see Chapter 12). However, it is essentially still a color histogram.

The second color feature to mention extracts the *dominant colors* of a media object. Dominant colors are those that are present in many pixels and arranged in cohesive blocks. Semantically, dominant colors are related to the loudness of audio. The short-time energy feature transform, for example, is only relevant where a significant difference in loudness between different sounds appears. Likewise, the dominant color feature transform is only relevant where some colors are of outstanding importance.

As suggested in Section 3.4 a simple form of this feature transformation can be based on the color histogram by selecting the three colors with the highest histogram entries as the dominant colors. This approach, however, does not take the cohesion of pixels into account, even though cohesion is of paramount importance for human perception of dominant colors. An advanced approach will, therefore, take the object structure into account – for example, in the following way:

1. Initialize an empty color histogram
2. Perform a color-based segmentation of the input object
3. For each cohesive region:
   (a) Compute the average color
   (b) Count the number of pixels
   (c) Add the squared sum of pixels for the average color to the color histogram
4. Select the three largest bins as dominant colors

In this approach, colors are associated with regions and the influence of great regions is emphasized by the square function. Therefore, the dominant colors will be those that appear in large unicolor regions. Color-based segmentation, by the way, is discussed in the last section of this chapter.

The MPEG-7 standard provides a dominant color feature transformation as well. This function computes the percentage of pixels of a particular color as well as the variance with respect to the average of each color. The results are quantized to n bins.

The MPEG-7 standard suggests two more color feature transformations: *color layout* and *color structure*. Color layout first applies a localization function by segmenting the input object into 64 equally sized areas. For each area, the average color is computed. Eventually, the result is transformed by the two-dimensional cosine transform. Color layout is, therefore, a spectral feature – see Chapter 13 for more. The result is only nominally a color description. In his earlier work, the author could show that, actually, color layout is a splendid texture feature [85].

The color structure feature transformation employs a special color model named *HMMD* that is actually related to the HSV model. Since it has four channels the HMMD color model is in comparison to the CIE model very redundant. Color structure first transforms the input object into the HMMD space. Then a color histogram of 184 colors is populated by counting color averages over a moving grid of pixel locations. The resulting description is very sensitive to the distribution of colors. In particular, colors evenly distributed over large areas of the media object will be strongly represented in the color structure description.

The latter two feature transformations are examples for the attempt to localize color information. Color histograms and dominant color descriptions that are extracted from entire media objects are called *global* descriptions. Global information is of interest if the media objects under consideration show simple compositions, i.e. if they do not contain complex objects and object relationships. Since most practically relevant scenarios are not based on simply structured material global color information is only of limited use. On the other hand, local descriptions such as, for example, the color histogram of a particular element in a media object can be very effective in describing the information in a media object. Since color was one of the first cues to be investigated in visual media understanding, it was attempted to solve the problems of localization and color description together. The result were feature transformations like color layout and color structure. Modern feature transformations, however, separate the localization problem from the color description problem. Approaches for localization are discussed in the last section of this chapter and in Chapter 14.

The MPEG-7 standard for content description was a major step forward for visual media understanding since for the first time it defined content-based fea-

ture transformations for image and video material. Shortly after release of the standard the author investigated the data quality of the resulting descriptions. As already mentioned above, he could show that some feature transformations do not provide the type of description intended while others are highly redundant. See [85] for details. However, despite their shortcomings, the color feature transformations of the MPEG-7 standard are today popular choices for many media understanding problems.

In summary, the two major color descriptions are the color histogram – either global or somehow localized – and dominant colors. Investigated by the raster of the fundamental problems of media understanding these color descriptions have in advantage that they are often of low dimensionality, that their extraction does not require a large set of parameters, that it be performed efficiently and that noise and distortions do not play an important role in the extraction process. On the other hand, the gap between a color histogram and the semantics of the content is usually significant. Methods to bridge this gap include localization on the object level and joint usage of color descriptions with other visual descriptions, for example, texture descriptions.

5.3 Texture Description

What is texture? One possible answer would be that texture is the statistical description of the visual sensation created by light reflected from the surface of some scene – hard to understand, but the expert will still find it unsatisfactory. A better way is to look at examples. Figure 5.6 shows some from the Brodatz data set commonly used for the evaluation of texture feature extraction methods. The leftmost pair of images shows the difference between a fine and a coarse texture, referred to as *coarseness*. The second pair differentiates between regular and irregular textures (*regularity*). The last three refer to the *directionality* of textures: the first image is vertically structured, the second one diagonally and the third one is non-directional.

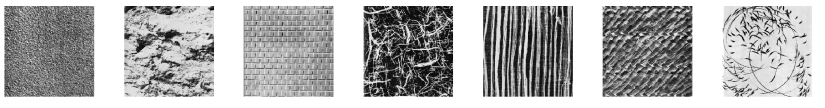


Figure 5.6: Brodatz Textures.

Coarseness, regularity and directionality are three very important properties of textures. It is, therefore, the goal of visual feature extraction to represent these three properties in descriptions. The methods introduced in this section

were designed for this purpose. Before we do that, however, a few remarks have to be made. Firstly, like the audio and color feature transformations discussed above, texture feature transformations employ the same building blocks, namely localization, quantization and aggregation. We will make this transparent in examples below.

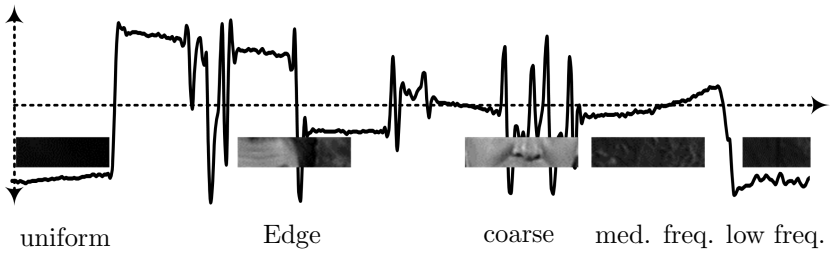


Figure 5.7: One- and Two-Dimensional Texture Examples.

Secondly, practice is unlike theory – also for textures. Figure 5.7 illustrates some textures extracted from the leading example together with the associated waveform. None of these textures has a clear direction. The coarseness is disputable and regularity degenerates partly to uniformness. The waveforms show that what we call a distinct texture is actually mostly a derivation of small magnitude. The only exceptions are edges that create an immediately recognizable wave pattern. Hence, what we are mostly looking for in texture feature transformation are delicate fluctuations of the signal that can – unfortunately – easily be confused with noise.

Thirdly, we would like to point out the semantic similarity between visual texture extraction and rhythm detection in the audio domain. Both concepts are best described in statistical terms as the precision of appearance of recurring patterns. Like a rhythm has a beat pattern with characteristic amplitude peaks, a texture has a basic pattern with characteristic luminance peaks that reappears – slightly varied – over space like the rhythm pattern reappears over time. The fundamental problems of texture recognition are, therefore, the same as for rhythm detection:

- Identification and description of the basic pattern
- Tracing of the fundamental pattern over space
- Description of the trace in terms of variance

It is not surprising that the methods employed to solve this problem resemble

those employed on rhythm detection.[1] Below, we discuss simple statistical models as well as autocorrelation and the application of density estimation methods. All of these methods have in common that they neglect color information. We assume that the visual stimuli are given gray-scaled, for example, transformed as introduced in the last section.

Given a gray-scaled image x, i.e. a matrix of cardinal numbers, some simple statistical moments of the texture can be extracted by the formulae given below. The measure x_e expresses the energy represented in the image. If combined with localization (e.g. to 16x16 subimages) and aggregation this measure is equivalent to the short-time energy extracted from audio samples. The values x_m and x_s are just the first two statistical moments mean and variance.

$$x_e = \sum_{l \in L} x(l)^2 \tag{5.3}$$

$$x_m = \frac{\sum_{l \in L} x(l)}{|L|} \tag{5.4}$$

$$x_s = \sqrt{\frac{\sum_{l \in L} (x(l) - x_m)^2}{|L|}} \tag{5.5}$$

These measures do not yet measure coarseness, regularity and directionality. For this purpose, we suggest the following approaches.

- Coarseness can easily be estimated by comparing the statistical moments for the given visual stimulus at the full resolution with the same moments at reduced resolutions. The longer the moments remain constant at downsampling the coarser the texture is.
- Regularity can be measured by localization and aggregation. If the statistical moments remain similar in, say, 16x16 subimages of the original stimulus the texture may be considered regular. If the variance of the moments is significantly above zero, it may be considered irregular.
- Directionality can be measured by employing the regularity algorithm on specific directions. Vertical directionality can, for example, be detected by computing the variance of the statistical moments for vertical neighborhoods (columns of subimages). If the variance is close to zero, we may assume vertical directionality.

[1] And the other way around! Most of what we discuss here has a perspective to be employed on audio as well.

These moment-based feature transformations are just examples. Others do exist. For example, the *Tamura features* were very popular during the 1990ies [363].

Another form of statistical description of textures is interpreting them as Markov Random Fields (MRF), i.e. as nets where each pixel is a *state* and luminance changes are regarded as *probabilistic state changes*. In consequence, the input stimulus can be employed – as any MRF – in order to train a density of conditional probabilities (state changes), for example, using Gibbs sampling. The resulting density function is a description for the texture of the input stimulus. See Chapter 9 for more details on probabilistic models and density estimation.

Above, we stress the similarity of rhythm detection and texture description. It is, therefore, not surprising that autocorrelation – of paramount importance in linear predictive coding – is also a strong method in texture description. Given a framework of variable-sized windowing and a convolution operator, coarseness, regularity and directionality can be described as follows:

- Coarseness can be measured by computing a correlogram for varying window sizes. The level with the highest autocorrelation is a measure for the coarseness: The higher the level, the coarser the stimulus.
- Regularity can, likewise, be measured by a correlogram simply by detecting the peaks in the correlogram. If few high peaks exist, the texture may be regarded as regular. Otherwise the stimulus may be regarded an irregular random signal.
- Directionality can be measured by computing correlograms for the directions of interest – like in the approach based on statistical moments. If peaks exist, the texture may be regarded as directional.

One particularly interesting holistic texture model based on autocorrelation is the *simultaneous auto-regressive* (SAR) model. In the SAR model, the gray values of the input stimulus x are transformed as follows:

$$x(l) = \sum_{y \in \theta(x,l,L_{moore})} w(y).x(y) + n(l) \tag{5.6}$$

Here, y are the neighboring locations of location l. The function $w(l)$ weights neighboring gray-values and can be used to make the model sensitive to particular directions by defining symmetric waves perpendicular to the direction of interest. The last term is a Gaussian noise term optimized from training data (e.g. by squared error minimization). The SAR model is auto-regressive. The process has to be repeated layer-wise until all output values $x(l)$ are stable. Then, the entire output is a description for the texture of the input image. That is, different texture characteristics cause different SAR models. The SAR model

can be extended by applying it at different scales and on rotation-invariant stimuli. Scale invariance and rotation invariance are discussed in Chapters 14 and 12, respectively.

The MPEG-7 standard defines two texture feature transformations: *homogeneous texture* and *texture browsing*. The first rotates the input stimulus in steps of 30 degrees and applies a two-dimensional Gabor wavelet transformation (see Chapter 12). The coefficients are cleverly quantized in order to provide an expressive description. This approach to texture description follows the idea that the coefficients of integral transforms express a texture-like property. Above we declared that MPEG-7 color layout acts statistically more like a texture feature transformation. This is due to the application of a sine-based integral transformation on pixel values. MPEG-7 texture browsing employs a scheme similar to homogeneous texture, but condenses the description to just twelve bits describing coarseness, regularity and directionality in four bits each.

In summary, texture description can be based on a variety of methods. We discussed statistical moments, joint densities, autocorrelation and integral transforms. In fact, many more approaches were developed – in particular, before the Millennium. Today, the MPEG-7 methods appear to be the most popular though, for example, the SAR model has proven very effective in the past. The introduced statistical models can be implemented efficiently and are, therefore, of interest for applications with limited resources (mobile setups) or broadband data (video).

5.4 Description by Shapes and Spatial Relationships

Shape is the third major visual cue next to color and texture. In this section, we define the shape of an object as its contour, i.e. the contrast between the object border and the object background. Obviously, in complex scenes, occlusions occur and the point of view causes distortions of the shape. In media understanding, these problems are considered types of noise that influence the shape description negatively. The elimination of this type of noise goes far beyond the methodology of media understanding. Such methods are developed in computer vision (see, for example, [348]). Therefore, we limit ourselves to the as good as possible extraction of the visual part of the shape border without the correction of distortions.

The edge perception of primates has been investigated since the 1960ies [164]. Figure 5.8 illustrates two major findings. After stimulation of the receptor cells, one of the first operations is to distinguish edges, i.e. dark/light contrast. Edges are distinguished by their *angle* and by their *length*. The length is at least categorized in two classes: short edges that end within the field of view and long

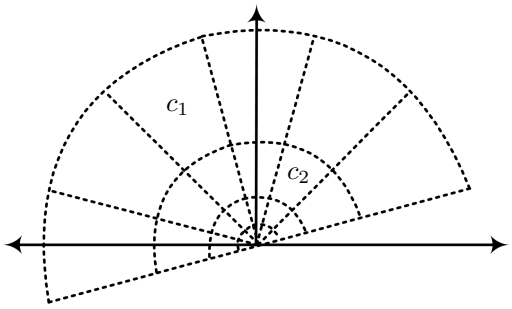


Figure 5.8: Human Edge Perception and Categorization.

edges that do not. In later processing steps, edge lengths are categorized by their length at a fixed step width. Human visual perception is largely based on this stream of edge information. These results are interesting for us, since the shape description is, likewise, based on edge detection. The cognitive findings justify the methods introduced below, because human visual perception is an extraordinarily powerful tool.

All feature transformations introduced in this section are based on edge extraction. Hence, we will discuss this problem first. Then, we explain transformations based on statistical moments, aggregation and other methods.

The purpose of edge extraction algorithms is to identify all points in a visual stimulus that belong to the border of an object, i.e. a contour with significant contrast. Since the 1960ies, a variety of methods have been proposed to solve this problem. Splitting and merging of similarly colored groups of pixels is one approach. Eliminating all pixels below a certain luminance threshold is another. One very powerful approach is based on correlation of the gray-scaled input image with pre-defined edge operators. Typically, such operators are 3x3 matrices with, for example, the following content:

$$o_{sobv} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}; o_{sobh} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}; o_{lap} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

The first two matrices define the *Sobel operator*. The third defines the *Laplace operator*. The first matrix is sensitive to vertical edges, the second to horizontal ones and the third to isolated points. See Chapter 14 for applications of the Laplace operator. A typical edge segmentation algorithm based on the Sobel operator can be defined as follows:

1. Convert the input stimulus to gray scale

2. For every non-border pixel do:

   (a) Convolute the L_{moore} neighborhood with the two Sobel matrices: $y = o \otimes L_{\text{moore}}$

   (b) If the convolution sum exceeds a pre-defined threshold replace the pixel value with value '1' otherwise '0'

The result is a binary image of non-zero edge points. Typically, the result is refined by eliminating all isolated points as well as very short edges, etc. Possibly, the algorithm can be employed on different scales. Then, only those locations are considered edges that are non-zero on $n\%$ of all scales – which is an application of *belief*.

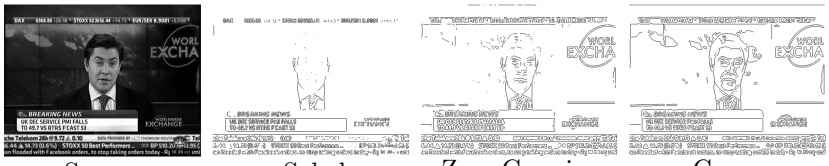


Figure 5.9: Comparison of Edge Operators (© *CNBC*).

Figure 5.9 compares a few edge extraction algorithms based on this scheme. The leftmost image shows the source, the second the output of the Sobel operator. The third algorithm considers edges at zero crossings of the image information while the rightmost image shows the output of the *canny edge operator*, an algorithm that performs several optimization tasks along with the correlation procedure. As can be seen the canny edge operator deserves its name, it produces the best results. It requires, however, also the most resources and is, therefore, only applicable under ideal circumstances. The Sobel operator is usually considered to provide the best trade-off between quality of the result and execution time.

Once the *edge map* has been extracted, it can be used to either directly generate a shape description or to segment the input image. An example for the first option is the MPEG-7 *edge histogram* feature transformation. This algorithm employs simple 2x2 operators in order to extract horizontal, vertical, diagonal (two types) and non-directional edges. In the second step, each point is added to a histogram with five bins depending on whether or not it belongs to an edge of the corresponding category. The edge histogram is typically localized to 4x4 subimages and computed on the level of macroblocks. In the above-mentioned statistical evaluations of the MPEG-7 standard, the author could show that the edge histogram provides very useful descriptions for all sorts of visual material.

A similar approach to the edge histogram would be the application of the Hough transform that is discussed in Chapter 12.

Figure 5.10: Principle of Energy-Based Contour Models (© *CNBC*).

The edge map can be employed for object segmentation in various ways. One simple approach is to focus on closed lines in the object map only, remove all other edges, fill the remaining areas and perform a logical AND-operation with the source image, which results in the segments of the original image. Additionally, almost-closed contours can be completed by some image enhancement algorithm.

Another more sophisticated group of approaches for image segmentation that are based on the edge map are *energy models*, for example, the *active contour model*. Here, the idea is to transform the edge map into a parameterized model that can be employed on the source image for object segmentation. All energy models try to minimize the opposing forces present in the model. Figure 5.10 shows an example where the model is a circle. In the active contour model, the two major forces are:

- The degree of fit of the model. In the figure, for example, the dotted line fits better to the object than the dashed line.
- The degree of deformation of the model. In the figure, the dashed line is less deformed than the dotted line.

The optimization process of the energy model tries to find an equilibrium between these forces. The resulting model can be employed for object segmentation or directly as a contour-based object description.

If the edge map or some model is used in order to extract an object from the input stimulus, we have again two options. The segmented object can be used directly as a description – like a visual keyword, or another feature transformation can be applied in order to arrive at an invariant shape description. For this purpose, shape moments and statistical moments are frequently used. Typically employed shape moments include:

- The size and aspect ratio of the minimum bounding rectangle around the object.
- The length of the border of the object.
- The ratio of object size and border length as a measure of its circularity. The larger the ratio the more circular an object is.
- The number of holes in the object, i.e. the number of smaller objects fully embedded in the object

Many more shape moments do exist, but these are the most frequently used ones. Typically employed statistical moments are the moments of first, second and third order: mean, standard deviation, skewness. However, tailor-made sets of moments exist that outperform the standard moments. One example for these sets is the set of *Hu moments*. For an object o the first four Hu moments $\mu_1 - \mu_4$ can be defined based on the central moments $\alpha_{p,q}$ as follows:

$$\alpha_{p,q} = \sum_{l \in L(o)} (l(x) - l_c(x))^p . (l(y) - l_c(y))^q \tag{5.7}$$

$$\mu_1 = \alpha_{2,0} + \alpha_{0,2} \tag{5.8}$$

$$\mu_2 = (\alpha_{2,0} - \alpha_{0,2})^2 + 4\alpha_{1,1}^2 \tag{5.9}$$

$$\mu_3 = (\alpha_{3,0} - 3\alpha_{1,2})^2 + (\alpha_{0,3} - 3\alpha_{2,1})^2 \tag{5.10}$$

$$\mu_4 = (\alpha_{3,0} + \alpha_{1,2})^2 + (\alpha_{0,3} + \alpha_{2,1})^2 \tag{5.11}$$

Here, $l(x)$ represents the x component of location l and l_c is the location of the center of gravity in the visual object. In total, seven Hu moments do exist. However, the first four already provide a fair object description.

Alternative to the description by moments, object segments can be described by the coefficients of some transformation. The MPEG-7 *region-based shape descriptor* employs the Angular Radial Transform (ART, see Chapter 12). The ART defines a set of 32 two-dimensional circular sinusoid objects that are each convoluted over the input image. The resulting coefficients constitute a very useful object description. The ART is particularly successful in the recognition of quasi-circular objects, e.g. faces.

A typical description of all shapes will be composed of some set of shape moments for the n largest objects. The number of objects present in a scene may vary widely, but as stated above, descriptions must be of fixed length in order to be utilizable for efficient categorization. However, the categorization need not only be based on the object content but also on the spatial relationships of the objects. One simple approach is the so-called *2D string*. The basic idea is

very simple: A 2D string is a symbolic representation of object relationships in horizontal and vertical direction with respect to some origin.

An example makes it clear. Suppose we have identified four objects in a face image: left eye (A), right eye (B), nose (C) and mouth (D). The 2D strings for these objects would be defined as follows:

```
u = A,CD,B
v = AB,C,D
```

If we consider the origin of the space in the upper left corner, the strings u, v (traditionally, used for x, y dimensions) express that object A comes horizontally before C and D, and C and D come before B. Vertically, A and B are located at the same position, then comes C and finally D. 2D strings express the spatial relationships of objects very efficiently. However, they require semantic labeling of the objects, otherwise the strings become arbitrary. The categorization of 2D strings can be performed like the categorization of any symbolic media objects.

In conclusion of this section, we would like to stress the importance of sophisticated object segmentation for visual media understanding. Object segmentation is the most important pre-processing step for the extraction of other descriptions such as color, texture and localized information. Without proper object segmentation, unrelated content may be mixed into one description, which is obviously not desirable. Object segmentation, however time-consuming, improves the quality of the media understanding process dramatically.

Chapter 6

Description of (Quasi-)Symbolic Media

Discusses the differences of symbolic and quantitative sample types, introduces the term quasi-symbolic, describes feature transformations for stock data, text and bioinformation, and identifies analogies between these methods.

6.1 Symbolic Media Types

In the last two chapters, we have introduced methods for feature extraction from audiovisual media objects. These types of media are based on quantitative samples. In this chapter, the situation is different. The media objects considered below use a symbolic carrier. It is, therefore, necessary to reflect (in this section) whether or not it makes sense to apply the concept of feature transformation on symbolic media objects. Since our answer is a clear yes, then, in the three following sections, we discuss feature transformations for stock data, text and bioinformation.

Even in the quantitative domain we have already encountered symbolic concepts. The 2D strings employed for shape description represent a symbolic description of the spatial relationships of visual objects. Of course, this concept could, likewise, be applied on any other type of media that may be composed – spatially, temporally or along some other set of dimensions – of events. The symbols of 2D string are the named events (e.g. shapes) encapsulated in media objects. In this sense, every template or model is actually a symbol in some sign language. The discussion thread leading from here to semiotics will be followed

in Chapter 22. The technical discussion of template-based event recognition is the topic of Chapter 24. Furthermore, we have already introduced the ADSR sound model. The sequence of attack, decay, sustain and release is a bridge for crossing the semantic gap from windows of quantitative samples to symbols of sound. The question now is, does it make sense to carry the process even further, that is to extract more specific descriptions from the symbolic representation?

As we discussed in Chapter 2, the major difference between media objects as aggregates of quantities and media objects composed of symbols is the neighborhood concept. In quantitative media objects, neighboring samples are related (redundant). That is, the local function is to some degree smooth (though, of course, there is no strict smoothness in the discrete domain). The crux now lies in 'to some degree,' an expression not yet precisely defined. If the proposed degree of smoothness is not met by a type of media, the samples are interpreted as symbolic otherwise quantitative. In consequence, every media type with a weak concept of neighborhood may be regarded symbolic – even though the basis of the samples may be measurements. In the next section, we will give a detailed investigation of one particular type of media – stock – that may be considered as symbolic as quantitative. We call this phenomenon *quasi-symbolic*.

The scale of quantitative, quasi-symbolic, symbolic is not just relevant for the description of media types but in fact, for all data objects produced in the media understanding process. Descriptions, for example, the zero crossings values of some piece of music, are rather quasi-symbolic than quantitative, since the meaning of neighborhood (extracted from neighboring windows) is limited. After categorization, the resulting predicates are certainly binary symbols expressing the membership relation to some semantic class. Hence, media understanding can also be defined as *a process away from quantitative towards symbolic representations of digital media.*

In the discussion of the properties of media types, we counted in favor of symbolic media that they are less prone to noise. That is generally the case, simply because the limited set of possible sample values acts like a multi-stable system (a function of many local optima), i.e. it is rather unlikely that a recording error leads to false categorization. It is much more likely that despite the noise, the correct local optimum (symbol) is associated with the measurement. However, this reasoning is not true for the products of media understanding, namely descriptions and predicates. The transformation and categorization process may convert the noisy measurements into even more noisy descriptions (by unintentional emphasizing of the noise component) and eventually, completely wrong predicates. Therefore, symbols created by a media understanding process have to be regarded with caution.

Two general approaches to feature transformation of symbolic media objects are thinkable:

- Lossless transformation
- Lossy transformation

Of course, both approaches are lossy in the sense that the generated descriptions contain fewer symbols than the original media objects. The distinction refers to the semantic content of the media objects. Two examples from the text domain should make the distinction clear. Considering the phrase 'Yesterday, the shares of IBM went up by ten points' a lossless transformation would be the elimination of the flexion of the verb. 'Yesterday, the shares of IBM go up by ten points' has the same semantic content as the original sentence. However, if the temporal adverb was removed from the sentence, less semantic content would remain. In 'The shares of IBM went up by ten points' we do not know if this event occurred yesterday, today, this week or within the last year. Therefore, this transformation has to be categorized as lossy.

When should we use lossless transformations, when lossy? Theoretically, the first form is always superior. Of course, we do not want to loose semantics in the feature transformation process. We rather want to free the semantics from all the irrelevant (with regard to some query) parts of the media object. Practically, this goal is hardly achievable. If a feature transformation is defined cautiously enough not to lose any semantic content, it will probably preserve large parts of the irrelevant content. Actually, this problem is based on the same trade-off as the problem lossy/lossless media compression: The more lossy the transformation the more efficient the description but the higher the risk of losing valuable information. Below, we introduce feature transformations that represent a reasonable – empirically proven – balance between efficiency and risk of loss.

The three remaining sections investigate one type of data each. In the next section, we discuss feature transformations for the quasi-symbolic nature of stock values. Then we move to the text domain. Eventually, we combine the methods introduced for stock values and text for the application on bioinformation.

6.2 Description of Stocks

Technical stock analysis has a clear goal: Prediction of the future. In contrast to all other media understanding problems considered in this book, we are not interested in indexing of the media content for content-based search but, exclusively, in the projection of past patterns into the future. Therefore, most

methods of stock analysis employ some kind of predictive coding (for example, linear regression, moving averages, etc.).

The fundamental problem of stock analysis is as clear as the goal. Stock signals are not smooth and barely predictable, in fact, close to a random walk, i.e. the values of the future hardly depend on the past. The typical model of the stock signal is a so-called Wiener process. In a Wiener process, each sample is the sum of two components. The first one is a weighted average of the last n samples (in the simplest case, the last one). The second component is a random number drawn from some normal distribution. The first component reflects the hope that the stock signal is indeed more than just a random walk while the second models all external and internal influences on the value of shares. That is, the component drawn from the normal distribution summarizes all pieces of information that trigger buying and selling operations. Compressing all information in just one value causes high polysemy, i.e. the sources causing a particular change are no longer recognizable. Therefore, polysemy is the one of the fundamental problems of media understanding that is most relevant in the stock signal domain.

Obviously, the random walk component of stock signals makes prediction hard to achieve. In fact, the predictability depends on the weight of this component relative to the average of the last n samples. The higher this weight is the harder the prediction is to achieve, or the smaller is the *belief* in any pattern that may reveal the future development. Hence, stock understanding makes sense in imperfect markets with limited information and few players. Markets close to perfection should maximize the random walk component and, therefore, render technical stock analysis impossible. Below, we introduce feature transformations for stock data that provide descriptions usable for powerful prediction of imperfect markets.

We consider stock signals quasi-symbolic. This judgment may appear surprising since stock data is usually visualized in graphs (though symbolic representations exist as well). Our reasoning is the following. Charts make sense if high-bandwidth data are available. In our context, if an analyst is provided with updates of stock values per minute (or, per second), the stock function becomes a smooth signal. Then, visualization and visual prediction make sense. However, if only low bandwidth data are available, say, one sample per day, the signal resembles a very crude Wiener process where visualizations are not of practical use anymore.

For an example, please refer to Figure 6.1. The first and second charts show (quasi-)symbolic data, namely some investment funds and the GenBank sample gene string introduced in Chapter 3. The latter signal was turned into a graph by associating the characters standing for the amino acids with particular delta values (changes of the signal). For example, we associated alanine (A) with a value of -0.024. That is, every occurrence of an A in the protein representation

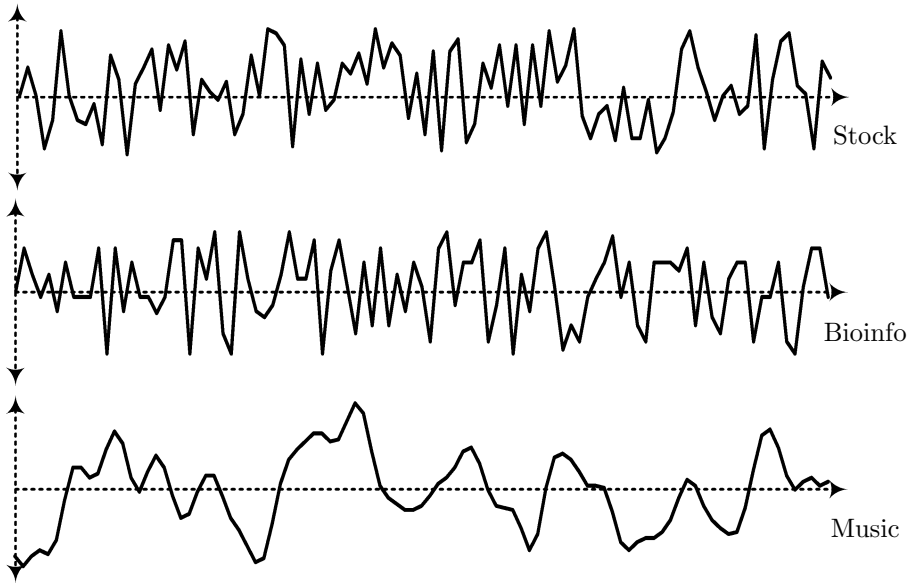


Figure 6.1: Delta Characteristics of Stock, Bioinformation and Music.

is visualized by a small decline. The actual values chosen for the individual acids are – since there is not a strong neighborhood relationship between amino acids – irrelevant. Important is only that each acid is associated with a distinct value and that all values are drawn from the same interval. The result is a typical random walk. The bioinformation chart does not reveal any interesting pattern (on first sight).

Surprisingly, the delta values of the stock data do not look much better. We used the day by day changes of some real investment funds to generate this chart. The result should reveal some pattern, but in the contrary looks very much like the bioinformation chart. The reason lies in the coarse sampling of stock data available to the public. One value per day is not much more than a random walk. In contrast, the third chart shows the delta values of some piece of music. The deltas are still close to a smooth signal and, therefore, express patterns that exist in the data.

The practical transformation of quantitative stock data to symbols is actually a media understanding process. In the first step, the stock values have to be described by delta values. In the second step, the delta values are categorized as

symbols. Various reasonable approaches are thinkable. We suggest the following algorithm:

1. Define an alphabet of n symbols.
2. Compute the delta values of all pairs of samples.
3. Build a density function (histogram) for the delta values.
4. Associate each symbol with an equal-likelihood interval of the density function.

Each delta value belongs to one particular symbol defined by an interval. With this algorithm, each symbol has the equal likelihood of appearance.

The following example shows a symbolic representation of the stock data visualized in Figure 6.1. We use the abbreviations used for amino acids in the genetic code as our alphabet. Using the algorithm above, the first values given in the figure equal this string:

```
KQKCFYKFEICSNAMQSHTPVENLINDGSKYYTCRGRHIRPSMYQTSFNESASVCERMT
HTFTHYNQYFDGHCLDDIADGSYQLFKMGHYANVYMKARNNYGAEVSWAVPGEEECFLH
VASDDEGSVDKYHKRCVFTIASNTNMMLTFPMKWDKLCIFCPSLMTNNAFQENGPKRSH
```

In comparison, the original gene string of the bioinformation curve given in the figure has the following form:

```
MTQLQISLLLTATISLLHLVVATPYEAYPIGKQYPPVARVNESFTFQISNDTYKSSVDK
TAQITYNCFDLPSWLSFDSSSRTFSGEPSSDLLSDANTTLYFNVILEGTDSADSTSLNN
TYQFVVTNRPSISLSSDFNLLALLKNYGYTNGKNALKLDPNEVFNVTFDRSMFTNEESI
```

For human cognition, again, no big difference is visible in the symbolic representations of stock and bioinformation. Of course, the distribution of symbols is different – but this piece of knowledge comes from the transformation algorithm, not from visible patterns.

Now, why should we prefer the symbolic description of stock data from the quantitative one? The three most important reasons are:

- The quantitative representation may mislead the analyst to approaching the prediction problem numerically – for which the data basis is insufficient.
- In the domains of text understanding and bioinformation analysis powerful methods for symbolic pattern analysis have been developed that can, with minor adaptations be applied on symbolic stock data.
- The principles of most of the quantitative methods applied on stock today can more efficiently be expressed in symbolic terms.

In conclusion, using a symbolic representation for stock data opens the door for new methods and speeds up the prediction process.

Still, the discussion of feature transformations for stock analysis that follows now will start with classic quantitative methods. Then, we will discuss how some of these methods can – more efficiently – be implemented in the symbolic domain. In combination with the remaining sections of this chapter and the chapters on the categorization of symbolic descriptions these descriptions provide a powerful toolbox for symbolic stock analysis.

Figure 6.2 visualizes some important descriptions used in technical stock analysis. The trendline in the upper part is the result of linear regression, i.e. the line minimizes the squared errors given the samples. The lower left part of the figure shows a moving average over the last ten samples.

Trendline and moving average are examples for statistical moments. That is, the entire pool of samples is condensed into a few representative values (descriptions). Many other such *financial ratios* do exist. The *advance decline value*, for example, is the ratio of the aggregated values of all rising and falling stocks on one day. The *relative strength* of a paper is the ratio of this paper compared to the market, etc. Eventually, the volume of a market is another interesting data series. From the trading activities, further moments (e.g. mean over time) can be drawn.

Generally, most other moments introduced for audiovisual description above could be used to describe quantitative stock. The short time energy, for example, should provide information similar to aggregated advance decline values. The zero crossings rate of the delta values resembles the so-called momentum of a market. Even logarithmic hearing (introduced in Section 4.1) has an equivalent in stock analysis. In markets with large deltas, the logarithm is used to scale the data to human understanding. In markets with small changes the exponential function (inverse logarithm) is employed to provide the opposite effect.

Returning to Figure 6.2, the lower central element is a so-called triangle, i.e. a pattern where ascent and decline are of equal size. The last two elements are resistance and support lines. The support line is a minimum not undercut over some amount of time. The resistance line marks a peak not exceeded for some amount of time. Many other similar patterns do exist (rectangles, flags, cups, etc.).

Such descriptions can efficiently be expressed in symbolic terms. Consider a simple alphabet of just three symbols: $\{a, b, c\}$ where a stands for ascent, c for descent and c for no change. Then, a small triangle can be expressed by the string *aaaaaccccc*. If we have one symbol per day (aggregations are simple) this triangle would span over ten days. A resistance line could be expressed by a maximum length pattern *aaaaaa* that may not be exceeded. Other descriptions could be provided in similar fashion. Such patterns, in particular, applied on a smoothed signal, simplify the identification of interesting structures (e.g. by

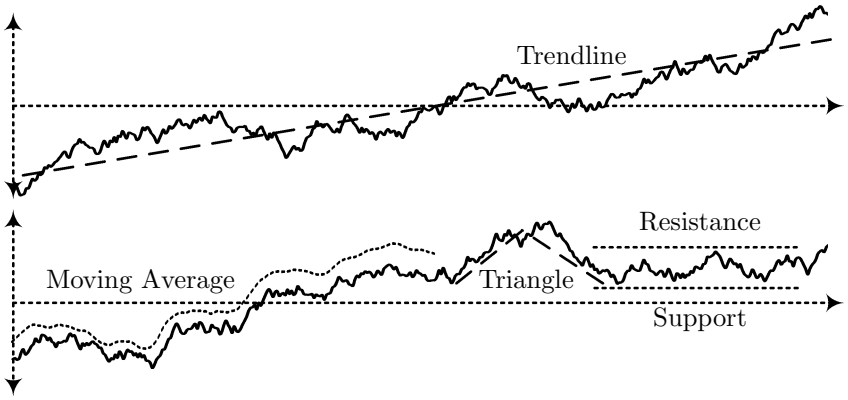


Figure 6.2: Features of Technical Chart Analysis.

regular expressions) dramatically.

However, describing the past is only half way towards prediction of the future. The actual prediction step requires some form of autocorrelation (like, for example, in linear audio prediction and visual texture recognition). That is, preceding patterns are mapped to the recent past and partial matches are computed along with some belief in the quality of each match. Since the past shape of patterns is known, patterns with a good partial match (high belief) allow a glimpse into the future. A simple quantitative implementation of this idea is the correlogram. By comparing windows of stock values by autocorrelation the most likely recurring time span can be identified. Since such rhythms exist in financial data only over (too) large cycles – we are interested in the immediate future – simple matching of repetitive symbolic patterns has the better prospect.

In conclusion of this section, stock is due to its characteristics a quasi-symbolic media type par excellence. Efficient quantitative description methods exist that suffer from the drawback that the narrow numerical basis is not sufficient for complex quantitative reasoning. Therefore, the belief scores of such methods are generally low. Luckily, many concepts of technical stock analysis can elegantly be expressed in symbol sequences – on which the methods of string transformation discussed in the next two sections can be applied.

6.3 Description of Text

Text understanding is a field of research with a long history, much longer than, for example, video understanding. In fact, automated text retrieval has been done for so many years that the name *information retrieval* – though sufficiently

generic to include retrieval of audiovisual and other media as well – is often set synonymous with text retrieval. Even though this fact is not always reflected, text retrieval follows the big picture of media understanding. On every level of text understanding (syllables, words, phrases, paragraphs, discourses, etc.) some feature transformation is performed followed by some categorization process. The feature transformation may be as simple as removing words to their principal parts. The result is still a description.

For the computer linguist, it may be unsatisfactory to mix problems as different as text retrieval and semantic text understanding. In many textbooks, these problems are discussed separately, each on the level of its description process and categorization process, often without differentiating between the two. Still, we follow the pattern introduced above and describe in this chapter only the feature transformation part of the different text understanding problems. The categorization is discussed – together with the categorization of all other media – in the subsequent chapters. Sticking to the scheme has the advantage of identifying similarities in the methods used and, possibly, learning from clever solutions employed elsewhere. With this potential at hand, we ask the text expert to be open-minded against our approach.

We summarize under the headline text understanding the methods employed in text retrieval and computer linguistics. The difference between the two domains is the *approach*, which is statistical in text retrieval and model-based in computer linguistics. We do not see a major difference in the goal, which is both times the understanding of the contents of text. However, in text retrieval this goal is approached with stochastic methods (e.g. n-grams of recurring word patterns) while in computer linguistics, the best-fitting model to some given source is of interest, with the reasoning that the meaning of the text will resemble the meaning of the best-fitting model. It goes without saying that the statistical approach has its limitations. Text semantics is only to some degree expressible in statistical terms. On the other hand, statistical methods are applicable on extensive text corpora – up to the entire world wide web. Statistics from such a large source may tell more about the semantics of human communication than any sophisticated model ever can. Furthermore, statistical descriptions can be of fixed size (e.g. word frequency histograms). In conclusion, both text retrieval and computer linguistics deal with text, follow the big picture of media understanding and are, in consequence, investigated jointly below.

Computer linguistics spans an umbrella over various goals of automated text understanding, including automatic summarization of text, the recognition of the meaning of the content (semantics), the understanding of the moods expressed by a text, simple copy detection and the recognition of authors by their style of writing. On the technical level, some important problems are morphological parsing of sentences, tagging, for example for spell-checking, and the understanding of discourses. In the latter case, the complexity comes primarily

from co-references (e.g. 'X did... He...').

All the above-mentioned problems are of interest to us. Beyond our interest lie general-purpose data mining, speech recognition (discussed in Chapter 25), handwriting recognition and sign recognition. The two latter domains are from the visual media understanding domain on which the same methods as on other visual media types can be applied.

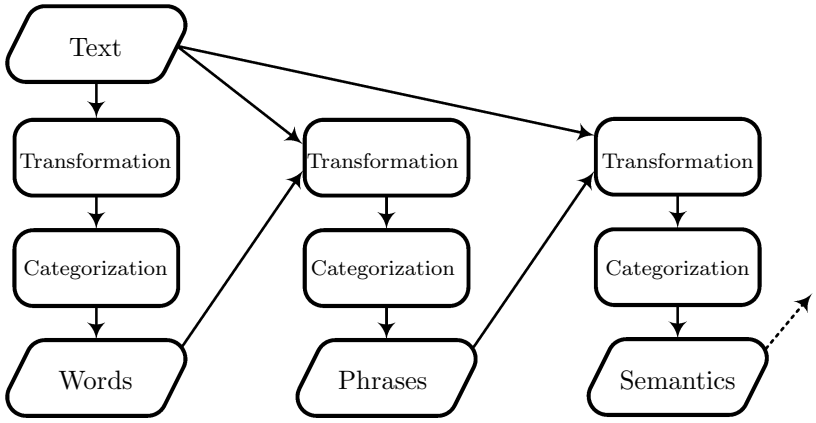


Figure 6.3: Text Understanding is a Cyclic Process.

The text understanding problems build on each other. It is hard to recognize sentences, if the words have not been properly recognized before. It is difficult to analyze a paragraph if it is not clear of what types of sentences in which order it is composed of. Therefore, text understanding is, maybe more than any other media understanding process, a cyclic process. Figure 6.3 illustrates this point. Given a text, in a first step words can be isolated.[1] Using the words (and, for example, throwing away a few of them) in a second step, phrases which serve as input for the semantic understanding can be categorized. The semantics could be used for further reasoning.

Why do we need to know the grammar of a sentence in order to be able to understand it? Simply because words may have different meanings depending on their position in the sentence. Hence, most sophisticated applications for semantic text understanding are based on parsing.

Before we start with introducing feature transformations, we would like to point out that we face three fundamental problems in text understanding: poly-

[1] Remark on the symbol system: We do not distinguish here between symbols representing syllables and symbols representing letters. The major difference is that, in the first case, some problems do not exist (e.g. detection of syllables) while others become due to the larger set of symbols a bit harder to model.

semy, curse of dimensionality and the semantic gap. The first problem becomes clear from the last paragraph. Words may have different meanings in varied contexts. A semantic application requires to understand these differences. The curse of dimensionality is a particular problem of text since, usually, every basic unit (syllable, word, sentence) is considered one dimension of the categorization problem. Since texts are, independently of the symbol system, constructed from many basic units, the description is almost ever high-dimensional. Eventually, the semantic gap problem is evident in text understanding, because writing is per se intended for storing semantics. Automated text understanding must, therefore, always deal with understanding semantics, something that need not be the case, for example, in audio event classification.

For the feature transformations discussed below, we define the following example paragraph – a business news item:

```
Shares of IBM went up by ten points. Nevertheless, big blue failed
to meet the expectations of analysts. Experts criticize the new
server generation as too far ahead of the customers' requirements.
```

One of the characteristics of text understanding is that feature transformation often requires additional input (next to the source document). Such input may be a dictionary, a *thesaurus* that organizes words by semantic similarity, or a model of the grammar. The media database from which the documents are drawn is usually referred to as *corpus*. The elements of sentences are frequently called *parts of speech* (POS).

One of the first tasks of text understanding is statistical aggregation. Words can be counted individually or in recurring phrases by so-called *n-grams*, i.e. specific word sequences frequently reappearing in the corpus. Such n-grams resemble histograms of audiovisual events with the difference that the types of events under investigation are usually not limited beforehand (unlike colors). A typical n-gram algorithm could have the following stages ($n = 3$):

1. For each group of three words (3-gram) do
   (a) Remove useless parts (see below)
   (b) Count the number of occurrences in the text
2. Count the mean of occurrences over all 3-grams
3. Use all 3-grams with, say, twice as many appearances as the mean as description

Simple word statistics such as word counting can be interpreted as moments of the text. In this sense, it may be reasonable to compute the variance of the

length of phrases as a description for the style of the author. In the example above, the short sentences may be characteristic of business news. This is, by the way, a good example for the principal property of media understanding that semantically very low feature transformations may be successfully used to explain semantically high level concepts (here, style of writing). Of course, very often we fail at the semantic gap.

Another important type of feature transformation is throwing away useless information. This step may be a prerequisite for more sophisticated moments but as well a feature transformation in its own right (for example, for tagging of words and phrases). Examples are the removal of the copula and the reduction to principal parts. If the time information is irrelevant, the example description may be reduced to the following one.

```
Share IBM go up ten points. big blue fail meet expectation
analysts. Expert criticize new server generation too far ahead
customer requirement.
```

In this example, we have removed words such as *of, the, by.* We have reduced plural to singular and the flection of verbs. The proper tools for these transformations are the grammatical flection rules and a thesaurus. In the next step n-grams may be computet for recurring phrases and patterns such as *share IBM, big blue, server generation* may be replaced by one name each. Of course, another reduction step would be replacing all uppercase letters by lowercase.

The identification of recurring patterns in text understanding is similar to object recognition in the audiovisual domain. In audio, we have already encountered the ADSR sound model (see Chapter 4). This model is a general pattern of sound. Likewise, we could employ a general pattern of a phrase such as *noun-verb-object* or, for fragments, *adjective-noun* or *adverb-verb.* The statistical identification of n-grams could be based on such text models very much alike the application of the ADSR model for sound identification. Morphological parsing is then just categorization of reduced text descriptions based on some text model.

The style of writing problem can be approached by autocorrelation. The n-gram implements already a simple form of autocorrelation. Recurring patterns are extracted from the text by simply counting all possible patterns of length n. For the detection of the style of writing n-grams can be further investigated by text models and weighted by their components. For example, an n-gram that contains a specific compound noun can be weighted as important. If such an n-gram reoccurs several times in a text, it may be characteristic for the style of writing of one author or one type of author. In the example above, the usage of the n-gram *big blue* may be characteristic for the endeavor of business journalists to make their texts more interesting by – literally – colorful synonyms. Hence, this string may be a good (characteristic) description of such a text.

On the other hand, *big blue* for *IBM* is an example for the hardest problem in discourse understanding: identifying co-references. Why are these two terms synonymous? Only because someone established this metaphor and over time it became common knowledge. Hence, the association between the words is more or less random. Still, a feature transformation that describes this relationship is thinkable. Based on a reduced phrase and n-grams of important POS the association could be identified from the usage as substantives and of alternated use of one of the two POS. This is a typical example why text understanding is actually media understanding of media understanding. The results of one description process are categorized and fed into another description process.

The same is true for the understanding of semantics and moods of text. Helpful descriptions include text summarization, n-grams, text models and statistical moments. The understanding itself, however, is primarily a problem of categorization. One algorithm for *describing* semantics and moods could be based on sets of similar words, for example, provided in the form of a thesaurus:

1. Remove all simple words from the text
2. Count the occurrence of the principal parts of all other words
3. Group and summarize the occurrences of similar words
4. Take the, say, five most frequent groups as a description of the content and/or the mood expressed by the text

This algorithm implements the similarity of words as some kind of neighborhood. Structurally, it resembles the dominant color algorithm: words are used like pixels, the thesaurus fulfils the role of the color model and the selection rule is both times a simple threshold. Many similar solutions for the understanding of semantics and moods are thinkable.

Technical remark: Text descriptions can be of *fixed* or *variable* length. The first type – preferred in media understanding – can, for example, be produced by a histogram of words, phrases, etc. that counts the number of occurrences per entry. The second – natural – case requires the application of an adapted classifier that matches descriptions properly and produces *semantic predicates* (e.g. 'positive mood=yes'). The semantic predicates can be fed into another media understanding process.

In conclusion, text understanding covers a number of diverse problems for which the most powerful feature transformations are removal of irrelevant elements, statistical moments and autocorrelation. In this section, we have not pointed out any starting points for the application of text methods on symbolic stock data, because text differs in one essential feature from symbolic stock data and bioinformation: One symbol alone hardly matters. In text, neighborhood

has a strong meaning in the sense that symbols are only informative in *con-text* (e.g. words, phrases). This may not be the case for symbolic stock data or bioinformation. The question of how to benefit from text understanding techniques in the understanding of symbolic stock data and bioinformation is discussed in the next section.

6.4 Description of Bioinformation

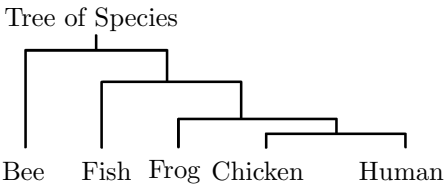


Figure 6.4: A Simple Gene-Based Taxonomy of Species.

Bioinformation understanding pursues the following major goals.

- *Sequence similarity*: Identification of the similarity of sequences of two gene strings. For example, often researchers are looking for the longest common subsequence of two gene strings.

- *Global and local alignment*: Alignment of two gene strings (global) or of essential elements of pairs of strings. One major problem here is the identification of anchors, i.e. equivalent substrings – used as starting points. Global alignment, can, for example, be used to build a *taxonomy of species* (Figure 6.4).

- *Multiple sequence alignment*: Practically, it is very important to perform local alignment for groups of strings (larger than two) in order to understand the complex relationships between genes better. Technically, multiple alignment is mostly a performance optimization problem.

- *Phylogeny reconstruction*: Identification of relationships between taxa (species) based on their genetic information. This is another way of measuring the similarity between gene strings.

- *Motif finding problem*: Which proteins follow from a particular DNA structure? The answer depends on so-called transcription factors, i.e. elements of the RNA.

- *Population genetics*: Members of a population carry mostly the same genetic information. All variations come from so-called single nucleotide polymorphism (SNP). Goal of population genetics is the identification of such SNPs.

In all of these applications, the major problems are the size of the data and, in consequence, the performance of the comparison process. State-of-the-art bioinformation understanding employs non-heuristic algorithms. For example, the sequence similarity problem is approached by the Needleman-Wunsch algorithm (equivalent to dynamic time warping), and an edit distance. See Chapter 8 for details on both methods. The algorithm guarantees the quality of an identified solution. Due to the large number of possible alignments, however, the central problem is the performance problem here. The same is true for most of the other standard solutions used in bioinformation understanding.

Our approach is fundamentally different, may be characterized as a heuristic and follows the big picture of media understanding. Rather than implementing a sophisticated matching process, we propose a cyclic media understanding process where complex gene data is reduced to descriptions, and descriptions are categorized by state-of-the-art machine learning techniques. If one iteration does not provide a satisfactory result, the categories, descriptions and raw data are fed into another cycle of media understanding, and so on. In this divide and conquer scheme, performance is not the big issue. Quality is – it depends on the quality of descriptions and the number of iterations. Below we do not investigate the standard algorithms of bioinformatics but focus on unorthodox feature transformations for the above-named problems. The classic way of bioinformation understanding is, for example, very well described in [359].

Before we begin with the discussion of feature transformations, we would like to give a few more technical details on the bioinformation domain (started in Section 3.4). The human genome consists of 23 DNA pairs (chromosomes) representing about 35000 genes. Of the 23 pairs 22 are non-sex and one is the sex chromosome. Every gene is composed of roughly 1000-2000 base pairs. The entire human genome is laid down in the GenBank database [276].

Figure 6.1 lists the so-called genetic code, a list of equivalences between amino acids and triplets of nucleotides. As can be seen, some acids are encoded by more than one triplet. Hence, a description of a gene string based on the symbol set of the amino acids is ambiguous. The same sequence of acids may be encoded by different nucleotides. On the other hand, the larger symbol set is roughly equivalent to the Latin character set and, therefore, more convenient to analyze for humans. Which encoding to use depends on the type of application. Implementing sequence alignment as a two-step process, first, raw alignment based on the amino acids and second, exact alignment based on the nucleotides would be a nice heuristic media understanding approach.

Amino acid	*Nucleotide triplets*
A	GCA, GCC, GCG, GCT
C	TGC, TGT
D	GAC, GAT
E	GAA, GAG
F	TTC, TTT
G	GGA, GGC, GGG, GGT
H	CAC, CAT
I	ATA, ATC, ATT
K	AAA, AAG
L	CTA, CTC, CTG, CTT, TTA, TTG
M (start codon)	ATG
N	AAC, AAT
P	CCA, CCC, CCG, CCT
Q	CAA, CAG
R	AGA, AGG, CGA, CGC, CGG, CGT
S	AGC, AGT, TCA, TCC, TCG, TCT
T	ACA, ACC, ACG, ACT
V	GTA, GTC, GTG, GTT
W	TGG
Y	TAC, TAT
End codons	TAA, TAG, TGA

Table 6.1: The Genetic Code.

Above we use the abbreviation RNA, which stands for ribonucleic acid – in contrast to deoxyribonucleic acid (DNA). RNA is very similar to DNA. The three differences are: RNA uses a different type of sugar (ribose instead of deoxyribose), instead of base thymine it uses uracil and may occur single-stranded while DNA is always double-stranded. RNA appears in various roles of which messenger RNA (mRNA) is the most interesting for us. For the creation of new cells, the DNA at the cell core is split into two independent strands and transcribed to mRNA. The mRNA is then translated to proteins, which build a fresh cell. This process is supported by other forms of RNA. The type of the new cell is determined by the motifs encoded in the genes. Since RNA is similar to DNA and used to transform different forms of DNA to the same cell structures, it is also called a *secondary structure*.

Above we named a number of structural alignment problems: local/sequences, global, multiple alignment. For all of these problems, we suggest a hierarchical processing model based on windowing.

1. Divide the source media objects in chunks of equal window length
2. Extract descriptions by some feature transformation
3. Establish correspondences between windows of different sources by categorization of the descriptions and correspondences identified earlier
4. Repeat the process with halved window size until the quality of the alignment does not improve any more

Suitable feature transformations could include throwing away all junk DNA, the transformation of the representation from nucleotide triplets to amino acids, n-grams of symbols that reoccur with high frequency (compare below, the identification of motifs), the transformation of amino acids to quantitative delta values and the computation of statistical moments, zero crossings, energy values, etc. of the numeric representations. Whatever turns out successful in experiments is a reasonable feature transformation. The categorization process could, for example, be based on simple distance measurement. See Chapter 8 for these and other possibilities.

Several remarks can be made on this approach. First of all, it is a divide and conquer strategy that has in advantage that it is simple and fast to compute. On the other hand, summarizing highly expressive gene codes into descriptions without reasonable explanation may appear insane to the domain expert. This proceeding, however, is a general approach of media understanding. Descriptions may not make any sense at all – as long as they provide the means to the categorization method to *differentiate between similar and unsimilar objects*. Another important remark is, that this is a localized approach. Hence, we encounter the – hopefully, already familiar – structure of localization, some kind of quantization and symmetric to localization, aggregation. Eventually, the iterative procedure allows to control the quality of the matching process. It is unreflected understanding in bioinformatics that alignments have to be perfect to the last bit. The algorithm is capable of reaching this goal but may also cut off at 90 per cent already – if desired. Hence, it introduces a new degree of freedom into genetic research.

For phylogeny reconstruction, one standard approach is based on the minimal number of mutations required to transform one taxa into another, visualized in a so-called ultrametric (dendrogram, see Chapter 8). Alternatively, the two taxa of interest could be summarized by the methods listed above, and similarity could be expressed as Euclidean distance. In particular, a simple histogram of the frequencies of occurrence of the amino acids could be a characteristic description.

The motif finding problem is a typical problem of autocorrelation. The goal is to identify structures of 5-20 base pairs (the motif) in RNA strings. The motifs

are contained in so-called transcription factors bounding sites and distinguished by the fact that they occur multiple times. Hence, one straightforward approach would be to compute all n-grams of $n \in [5, 20]$ and to identify all exceptionally often occurring elements. The aggregated structure of one specific n-gram (varying n) would resemble a correlogram.

Eventually, the problem of population genetics can be approached by the heuristic alignment algorithm. We look for small differences in a largely aligned pair of strings. The algorithm is tailor-made for the indication of such locations. In fact, already after a few iterations of the algorithm starting points for the identification of SNP should become visible. SNP should lie at locations where the match of two (or more) windows is suboptimal. If such a non-perfect match remains over further iterations, linear search in this window could be used to identify the exact SNP location.

Descriptions of bioinformation are per se of variable length. They can be made static by the same methods as suggested for text. The simplest approach is building a binary histogram of known genes with '1' for present and '0' for missing genes in a particular string. Such predicates could be employed for further analysis by predicate-based methods.

Now, which similarities between the methods employed on text and bioinformation can be identified? Obviously, the media domain is considerably different. As emphasized above, individual symbols are not as important in text as in bioinformation. Still, we suggest heuristic algorithms for the solution of the bioinformatics problems. The feature transformation process is mostly based on statistical moments and autocorrelation by n-grams – though the 'gram' is a single symbol in the bioinformation domain. The essential difference between text understanding and bioinformation understanding is the processing model. Text is best processed linearly, while for bioinformation, it makes sense to use a divide and conquer strategy.

How can the methods applied on bioinformation be employed on symbolic stock data? First of all, we have to make clear that these two domains pursue completely different goals. While stock data analysis is after estimating the future from the past, bioinformation understanding is about identifying similarities between groups of objects. Still, some methods are also applicable on stock data. For example, the n-gram approach could be used in conjunction with pre-defined patterns in order to describe the frequency of occurrence of the patterns. Such information would be valuable for prognosis. Furthermore, it may make sense to align multiple stock data objects before analysis in order to increase the expressiveness of the source data. For this purpose, the introduced alignment procedure could be employed.

At this point we conclude three chapters of simple feature transformations for the description of media objects. We have introduced a number of quantitative and symbolic schemes for summarization, redundancy elimination and simplifi-

cation of media objects. The Chapters 8 and 9 will show how these descriptions can be employed to perform the categorization step of media understanding. However, before we dive into this matter we have to leave a few words on the merging and filtering of descriptions. The next chapter will show how descriptions can be merged in order to perform true multimedia understanding and – in this process – how the level of redundancy in the descriptions can be reduced to a minimum.

Chapter 7

Merging and Filtering of Descriptions

Introduces how descriptions are merged, the concept of redundancy, that redundancy is a bad thing, the various remedies provided by information filtering, factor analysis, visualization of descriptions and statistical testing.

7.1 Merging of Descriptions

The four sections of this chapter introduce fundamental concepts of information filtering for media understanding. The first section explains merging of descriptions extracted from technically different yet semantically related media objects. The next two sections deal with redundancy reduction of merged and unmerged descriptions. The last section discusses statistical testing as well as options for the visualization of descriptions that support the better understanding of the morphology of descriptions. Advanced information filtering methods are discussed in the second and third part (Chapters 16 and 26).

The major purpose of information filtering in media understanding is the elimination of undesired *redundancy* in the descriptions. Redundancy means that two or more elements (one quantity/one symbol) of a description express the same information, i.e. – in quantitative terms – have similar mean and variance. In this case, one element would be sufficient as a description. The other element could be discarded.

Theoretically, each feature transformation is supposed to produce descriptions that are without redundancy. In practice, none of the more than thirty

feature transformations that we introduced in the preceding chapters can provide redundancy-free descriptions for any type of media content. Furthermore, by gluing together the descriptions of one event captured in separate media channels/streams together, additional redundancy is being introduced. Redundant elements, next to consuming resources for no benefit, may have confusing effects on the categorization process. Hence, elimination of redundancy is desirable.

In this information filtering chapter we deal exclusively with quantitative media. Merging of symbolic descriptions will be discussed in Chapter 16. The reasons for postponing symbolic media are threefold:

- Quantitative description elements are usually modeled as being the sum of a signal component and a noise component. The noise component is drawn from the same distribution for all elements of the description. In consequence, it may cause redundancy between elements. This model provides a convenient explanation of redundancy in description elements. Symbolic media are not able to express such a noise component due to missing gradations. Hence, redundancy in symbolic descriptions can only be explained by *rhythms* of recurring symbols – something that we already covered in the last chapter.[1]

- Merging of descriptions of one event (e.g. the audio and video track of a surveillance video showing an accident) requires that descriptions are of fixed length, because similarity measurement between events has to be done element-wise. If descriptions could be longer or shorter depending on the media content events would not be comparable in the simple element to element form anymore (see below for more). Symbolic descriptions are per se not of fixed but variable length. Of course, symbolic media can be expressed statically (e.g. word/gene histograms) but this is not the standard. Merging – the most important source of redundancy – can therefore, not be done in the standard way. Hence, symbolic descriptions are generally less interesting for information filtering than quantitative descriptions.

- Eventually, most information filtering techniques employ statistical algorithms and are, therefore, strictly numeric. For symbolic data, only few methods like Huffman coding are available.

In the remainder of this section we discuss the merging problem. We introduce the general concept, the properties of the resulting feature space, operations

[1] There is, however, a general-purpose approach to redundancy elimination for symbolic descriptions: zipping them (lossless compression). Compression methods like Huffman coding eliminate the redundancy of symbolic descriptions. Unfortunately, at the same time zipping destroys the semantic content of the description. It is, therefore, not applicable to media understanding.

performed alongside and special issues of multimedia description merging. The statistical analysis of merged descriptions is explained in the subsequent sections.

Why should we want to merge descriptions in media understanding? Often, one event is expressed in several media objects or multiple channels of one media object. For example, a goal in soccer is perceivable visually (the round thing goes through the rectangular thing, as Sepp Herberger, German world cup coach 1954 put it) as well as audibly (people cheer). Exploiting all available channels makes the categorization process more reliable. In the example, if the round thing goes through the rectangular thing but people do not cheer then the goal was probably not counted (or nobody came to watch the game). Hence, the *fusion of descriptions* is a very important step for successful media understanding.

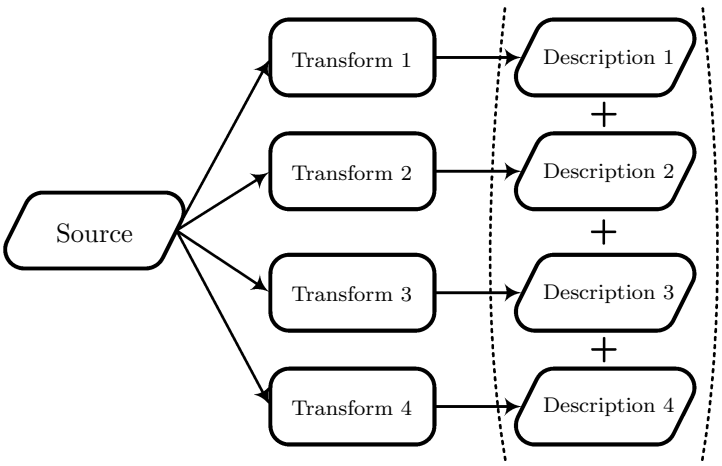


Figure 7.1: Merging of Descriptions.

Figure 7.1 illustrates the straightforward description merging process of media understanding. The source is a multimedia object with n channels. In the first step, all kinds of feature transformations are applied and descriptions are extracted. These descriptions (each one being a vector of numbers) are merged by simply concatenating them along the vector dimension. The result is a vector that may practically have ten thousand elements or more. This scheme has two prerequisites:

1. Description lengths are fixed, i.e. independent of media length or media content. Otherwise, positions of description elements would vary in the merged description string from the first element of the second description on.

2. The actual order of the descriptions in the merging process is irrelevant, but it has to be the same for all members of the population (e.g. all scenes of shots on goal in a soccer movie database).

These conditions are required by the categorization step of media understanding. Whatever form of categorization is used, at the center lies some kind of comparison process of descriptions. This comparison process relies on the rule that *specific description elements located in particular positions of the description vector have a particular meaning.* The merging process must not violate this condition.

There is no fundamental problem in concatenating the descriptions of one event extracted from diverse media sources. An art gallery application, for example, that desires to measure the excitement of a visitor that wanders through the exhibition, may use visual sensors for capturing the facial expressions, audible sensors for capturing comments and a brain computer interface for capturing brain activity. These modalities can easily be merged under one condition: the temporal context must be preserved. That is, the descriptions to be merged must have been extracted from the same window of time. Space may vary widely. Time is the glue that holds the elements of an event together. Of course, temporal gaps may occur where semantically justified (e.g. lightning and thunder in thunder storm events).

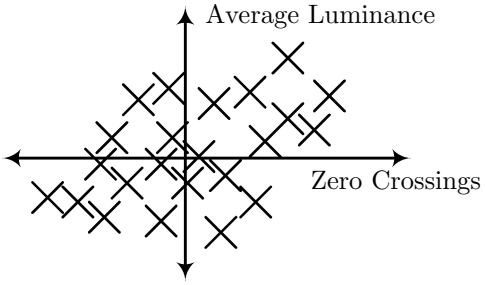


Figure 7.2: A Two-Dimensional Feature Space.

We said that the result of merging, the descriptions of one media event, is a (usually, long) vector. Media understanding is always performed on a database of media objects. For information filtering it is beneficial to merge the vectors of all media objects into one object, referred to as *feature space* (description space). Feature space is a matrix of description elements and media objects. We already encountered this concept in Section 3.2. Typically (not necessarily), the rows express the events while the columns express the descriptions. In terms of our mathematical notation feature space is just another array with a two-dimensional locations set.

$$F = [s_l | s \in S \wedge l \in L^2] \tag{7.1}$$

Figure 7.2 gives an example of an exceptionally low-dimensional feature space. The location set has a dimensionality of 23 media objects with two description elements each, i.e. $L = \{(1,1),(1,2),(2,1),(2,2),(3,1),...,(23,2)\}$.

A number of data problems can be identified by analyzing feature space. The above-mentioned redundancy is recognizable by a constant relationship (linear or over-linear) of two or more description elements over large shares of the media database. Furthermore, ineffective feature transformations that generate the same descriptions for different types of content can be recognized as rows with low variance in the feature matrix. Undesired distributions of description values can be identified by building histograms of values over the rows of feature space.

Below, we introduce a number of methods that fulfill these and similar tasks. However, in this proceeding we have to be careful about the levels of measurement (scale types) on which particular description elements are expressed, because the type of scale determines the mathematical operations that may be performed in the analysis. Figure 7.3 summarizes the four major types of scales.

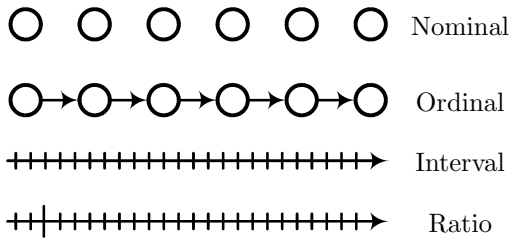


Figure 7.3: Levels of Measurement.

The simplest type of scale is the *nominal scale*, which is essentially a set of symbols without a general relationship. Symbolic descriptions are based on the nominal scale. Since not even addition and subtraction are defined on such data statistical analysis cannot be performed on nominal-scaled data.

The *ordinal scale* differs from the nominal scale in the one property that relationships such as *greater than* and *smaller than* are defined here. Hence, the symbols (often, numbers are used) can be *ordered* along the scale. However, arithmetical operations are not yet possible on ordinal-scaled data.

The *interval scale* extends the ordinal scale by defined intervals between each two symbols (then, always numbers). In consequence, addition, subtraction and multiplication by $n \geq 1$ can be performed. Most information filtering methods discussed in this book require a feature space where each row and column is interval-scaled. The quantitative descriptions extracted by the feature transformations discussed in the preceding chapters are mostly at least interval-scaled.

Ratio scale, eventually, extends interval scale by some origin (a zero value). If an origin exists, all arithmetic operations are performable. Ratio scales with a natural origin are often called *absolute scales*. One example would be the Kelvin temperature scale where zero is the absolute end of atomic movement.

It is of high importance for many methods of media understanding to take care that all description data are at least on the interval level. However, practically, this is not always the case. Since media understanding is an empirical discipline, the standard remedy to this problem is to simply assume the entire feature space to be interval-scaled, believing less in the results of analysis and performing more types of analysis in order to level the disbelief out. Moreover, the differentiation between nominal/ordinal and interval/ratio scales illuminates the gap between quantities (such as samples, descriptions) and symbols (such as characters, predicates). The practical difference between the two types may be bridged by some interpolation method, yet the mathematical difference is fundamental.

Continuing with practical problems, it is often helpful to normalize all rows and/or columns of feature space to the same range of values and/or statistical moments. For example, comparing two events by the mean of their description elements makes only sense if all description elements measure on the same range of values. If one description, say, average short time energy measures on the interval $[0, 8000]$ while the other, say, average luminance measures on the interval $[0, 255]$ then the latter would influence the mean less than the first. Such an effect can be avoided by normalizing all descriptions to the same range. The following method normalizes all elements of vector x to $[0, 1]$:

$$x(l) = \frac{x(l) - min(x)}{max(x) - min(x)} \tag{7.2}$$

The resulting values can be transformed to any desired range by adding an offset and/or multiplying with a scale factor. Sometimes, instead of a specific range, it is desired that description elements have particular statistical moments, namely mean and variance. The following method normalizes all elements of vector x to $\bar{\mu}_x = 0$ and $\bar{\sigma}_x = 1$:

$$x(l) = \frac{x(l) - \mu_x}{\sigma_x} \tag{7.3}$$

This normalization is, for example, used in factor analysis as a preprocessing step. The second normalization may, by the way, be used to replace the first, if the mean is chosen in the middle of the desired interval and the variance is chosen appropriately.

So far, we have dealt with constructing a feature space from individual media sources that represent one event. We have not considered multiple channels (e.g. audio channels) of the same event. Of course, every channel may be

considered an individual media object and treated accordingly. Very likely, this would practically lead to high redundancy of the merged descriptions. It is often superior to perform some combination operation instead. Frequently used methods are employing the averages of the corresponding description elements, the maximum, minimum or an otherwise – in some semantically reasonable sense – superior value. If it is not clear what method to chose it is advisable to try all options, apply the analysis methods suggested in the next two sections and choose the best-performing operation according to the filtering results.

In conclusion, the first step of information filtering is building a feature space. Merging descriptions has a negative effect on the dimensionality of the media understanding problem and, therefore, on the performance since it is inverse to a divide and conquer strategy. On the other hand, very useful normalization and analysis methods become applicable that allow to reduce the level of polysemy in the descriptions by eliminating noise, redundancy and other undesired components of the data. The next section will discuss simple analysis methods that can be applied on feature space.

7.2 Simple Statistical Filtering

This section summarizes methods for the more efficient representation of media events by descriptions in feature space. It starts with coarse and sparse representation, two methods for reducing the size and number of description elements. The discussion continues with statistical moments for the summarization of description elements, leading directly to regression and using differences instead of description elements. More complex methods for redundancy elimination are introduced in the next section.

In our mathematical notation, descriptions are arrays of samples s_l drawn from a sample set S. Aggregated in feature space, it makes a huge difference – for example, in terms of memory, but also in computation time – if the members of S are of type *integer* or of type *float*. Most feature extraction methods, however, neglect the problem of efficient storage and will rather provide floating point descriptions. *Coarse representation* is one method of information filtering for reducing the size of description elements. Coarse representation is one type of *source coding*.

The simplest form of coarse representation is reducing the precision, for example, by the *round*() function. Furthermore, unnecessary orders of magnitude can be eliminated by dividing by 10^x before rounding. More intelligently, the above-introduced normalization methods can be used for the same purpose. Removing a static offset can also be helpful to fit description elements into smaller types of data.

The process of coarse representation is straightforward. It is recommendable

to build a histogram (density) of all values of a description element over all media objects in feature space. Then, a proper reduction method can be chosen. If necessary, outliers can be eliminated or – by hand – represented by reserved values. The major difficulty in coarse representation is defining a transformation that is efficient but preserves the characteristics (in particular, distribution) of the data. The suggested techniques are harmless in this sense and recommendable for use.

Sparse representation goes one step further than coarse representation. Instead of increasing the density in the description elements, sparse representation techniques aim at making as many description elements irrelevant (e.g. zero) as possible. Sparse representation methods are similar to source separation algorithms (see Chapter 16). The general problem of representing a description x by weights w can be formulated as follows:

$$y = Ax \Rightarrow m_0(w) \rightarrow min \tag{7.4}$$

The goal is to identify a weight vector w with as few non-zero elements as possible. Measure m_0 counts the number of non-zero elements. Matrix A is a so-called *overcomplete dictionary* (or *codebook*). The columns of A are typical descriptions of typical media events. The weights w express the extent to which a particular description x is similar to the dictionary. Hence, sparse representation tries to represent descriptions by linear combinations of *prototypes*. The better the codebook the easier is the sparse representation.

Sparse representation has two obvious problems:

- Defining a good dictionary A
- Identifying the best vector w

The second problem is not a media understanding problem. Typically, a *matching pursuit* strategy or some other operations research method is employed. The dictionary can be defined randomly or by selected (typical) members of the feature matrix. If necessary, an iterative process of estimating or guessing A, computing w and checking its quality can be used for optimization (so-called expectation maximization, see Chapter 9).

Another form of sparse representation would be to replace n similar description elements by their statistical moments. The standard discrete moments of a description vector x of $n = size(x)$ are defined as follows:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}; v = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}; \sigma = \sqrt{v}; \kappa = \frac{\sum_{i=1}^{n} (x_i - \mu)^3}{n\sigma^3} \tag{7.5}$$

The moments from left to right are mean, variance, standard deviation and skewness. Mean and standard deviation are efficient ways of describing groups of description elements. We used them on a couple of occasions to define feature transformations. Skewness is very useful to find out if the distribution of some elements is skewed or not. Next to sparse representation, statistical moments can also be used for testing (see below). All moments are special cases of the following system:

$$\mu_{k,r} = E\big((x-r)^k\big) \tag{7.6}$$

Here, the moments μ at position r of order k are drawn from data x. If $r = 0, k = 1$ we receive the mean. If $r = \mu_{1,0}, k = 2$ we receive the standard deviation and so forth. For some media understanding problems it may be interesting to introduce higher-order moments as description elements.

Sometimes, however, the mean is not a good measure for a population. One example is the existence of a compact population with a few significant outliers. In this situation, it is recommended to use the median m instead of the mean, because it is less prone to outliers. One popular algorithm for computing m from a vector x is *mean shift*:

$$m = \frac{\sum k(x_i - m)x_i}{\sum k(x_i - m)} \tag{7.7}$$

Function k is a – typically Gaussian – weighting function. Starting from a predefined (e.g. random) m, the equation can be used iteratively to refine the mode until the level of improvement approaches some ϵ threshold ($\bar{m} - m < \epsilon$). Often, the algorithm is not applied on an entire description but just a neighborhood of some point. Mean shift is a simple and very popular algorithm in media understanding.

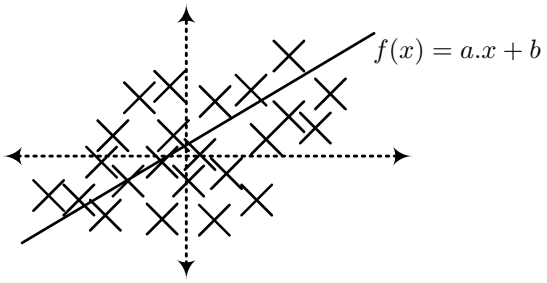


Figure 7.4: Linear Regression Example.

Regression is another simple model for the summarization of – here, longitudinal – data. If feature space is somehow stretched, i.e. has high covariance over two or more dimensions, it is a good idea to replace the individual description elements by regression parameters. Due to the form of computation, regression is also called the method of squared errors. Figure 7.4 illustrates the setting in the two-dimensional case. In a cloud of descriptions, we search for a regression hyperplane represented by parameters a, b. These parameters can be computed from a pair of descriptions (x, y) as follows:

$$b = \frac{\chi_{x,y}}{\sigma_x^2} \tag{7.8}$$

$$a = \mu_y - b\mu_x \tag{7.9}$$

$$\chi_{x,y} = \frac{\sum (x_i - \mu_x)(y_i - \mu_y)}{n} \tag{7.10}$$

Here, χ is the covariance of x, y. Traditionally, the correlation coefficient is employed as a quality measure for the representation:

$$\rho_{x,y} = \frac{\chi_{x,y}}{\sigma_x \sigma_y} \tag{7.11}$$

If the absolute value of the correlation coefficient approaches '1' the data is well represented by regression.

The last method for efficient representation to be mentioned here are *delta coefficients*. The idea is very simple. Instead of the description elements x_i we employ the differences between neighbors, and so on. With every iteration, one element is lost and the others can be represented by smaller types of data.

$$\delta = x_{i+1} - x_i; \delta\delta = \delta_{i+1} - \delta_i \tag{7.12}$$

This discrete scheme resembles derivation in continuous spaces. Figure 7.5 illustrates the effect. We suggested delta coefficients already for the description of stock data. Their application makes sense for any type of data where neighborhood has a (temporal or other) meaning, i.e. where difference is a semantic category. Delta coefficients are always useful if the primary interest is not in magnitudes but in (small) change. It is, therefore, not surprising that delta coefficients up to fourth order are employed in audio understanding, for example.

In this section, we introduced several simple models and algorithms for polishing feature space. Their effect is mostly positive in terms of performance, since they reduce the dimensionality of the data. On the other hand, valuable information can easily get lost by these methods which may increase the semantic gap between media content and descriptions. Since one of the biggest

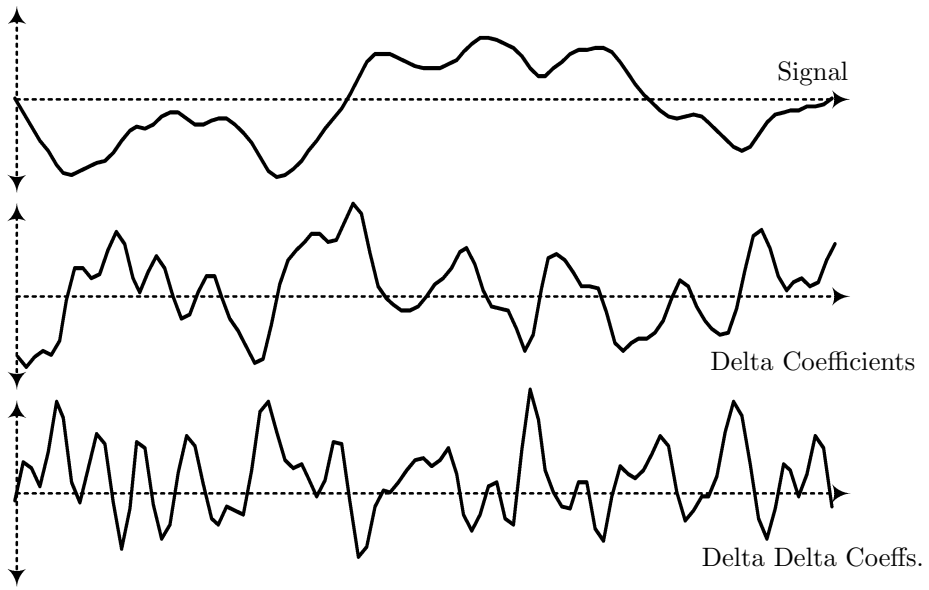


Figure 7.5: Signal, Delta and Delta Delta Coefficients.

problems of media understanding is dealing with large data sets the application of statistical filtering is still highly recommendable. However, more is possible – as we will see in the next section.

7.3 Factor Analysis

The methods discussed so far have in common that they seek optimization with limited information given. Some operate on individual description elements, others on groups. None utilizes the entire feature space. Hence, the gained benefit is limited. In this section, we discuss methods for redundancy elimination over the *entire* feature space: *factor analysis*, in particular, one method, *principal component analysis* (PCA). PCA is one of the most powerful yet elegantly simple methods for linear and non-linear redundancy detection and elimination in information filtering.

The fundamental hypothesis of PCA (we use the term synonymous with factor analysis, though several variations do exist and are used) is illustrated in Figure 7.6. The *variables* are the description elements representing *observations*

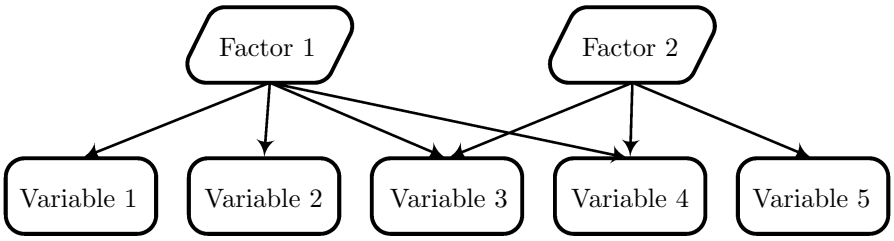


Figure 7.6: The Idea of Factor Analysis.

(another name for media events/objects/content) laid down in the *data matrix* (feature space). The factors are *linearly independent weight vectors* like those assumed for sparse representation above. The hypothesis is that the variables are just linear combinations of the factors. One factor contributes to the explanation of one or more variables, and each variable is explained by one or more factors. Redundancy is here defined as co-variance of variables and expressed in the weights of factors on variables. For some given feature space F the arrows in the figure correspond to W in the algebraic expression $F = W.X$. W is a matrix of linearly independent weight vectors called the *factors matrix*. X is the so-called loadings matrix – something very similar to a codebook.

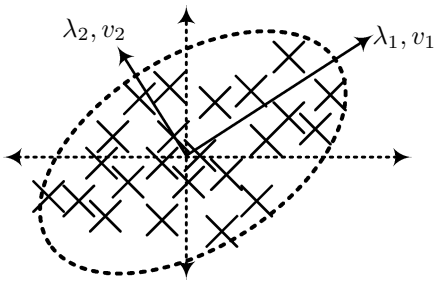


Figure 7.7: Eigenvectors and Eigenvalues in Factor Analysis.

Goal of factor analysis is to identify the loadings matrix X from the known F. If the loadings are known we can use the factors (computed by $W = F.X'$) instead of the redundant data. Since we will usually have much fewer factors than variables the result is large-scale redundancy elimination. However, the problem is obviously underdefined. The result depends, therefore, on the chosen algorithm and parameterization. PCA follows the strategy described in Figure 7.7. Feature space is viewed as a cloud of observations in some n-dimensional space of variables. PCA computes the elliptical hull of this *data cloud* and describes it by – necessarily, perpendicular – vectors pointing in the directions

of its major axes. The length of each vector represents the scatter of the data cloud in this direction, i.e. it is a measure of variance. The concatenated matrix of the first x dimensions is employed as the loadings matrix X.

This scheme may appear confusing at first sight, but is highly reasonable. What we have is a redundant linear space with covariances in some directions (in the figure, towards the upper right). What we are looking for is a transformed coordinate system where no covariances exist anymore. Hence, covariance (linear dependency) is synonymous for redundancy and transforming the base eliminates redundancy. The major axes of the elliptical hull are an orthogonal base of feature space.

The method that PCA uses for identifying the orthogonal base is as straightforward as beautiful. It relies on a given feature space F of n observations and m variables in which all description elements are normalized to zero mean and standard deviation one. The elements of feature space are assumed to be at least interval-scaled, i.e. subtraction and multiplication are possible. The algorithm consists of the following steps.

1. Compute the covariance matrix of F as $\chi = \frac{F'.F}{m-1}$. Subtraction of the mean and division by the standard deviation fall away due to the normalization.

2. Eliminate the factor matrix in the expression $F = W.X$ as follows: $\chi = \frac{F'.F}{m-1} = \frac{(W.X)'.W.X}{m-1} = \frac{X'.W'.W.X}{m-1}$. Since the components of W are linearly independent by definition we have $\frac{W'.W}{m-1} = I$, I being the unit matrix. In consequence, $\chi = X'X$. This result is called the *fundamental theorem* of factor analysis.

3. Compute the eigenvectors v_i from χ and use them as the columns of X. Furthermore, use the eigenvalues λ_i as measures for the importance of the eigenvectors.

The third step is a heuristic, yet a very good one. The eigenvalues and eigenvectors of some matrix A are defined in linear algebra as $A.v_i = \lambda_i.v_i$. That is, matrix A is just a weight of value λ_i in the direction of vector v_i. In other words, A is highly covariant in the direction of the eigenvector. Since all extracted eigenvectors are perpendicular to each other, they form a base naturally suited to the PCA problem. In matrix X the eigenvectors are sorted by descending eigenvalue.

This point could actually be the end of the story. Practically, however, and additional step is usually performed in factor analysis: *factor rotation*. The eigenvalues provide a good heuristic solution to the fundamental theorem – but not necessarily the best. Iterative rotation of X by some rotation matrix T with the property $T'T = I$ might cause further improvement. Various heuristic

rotation schemes do exist. One of the most popular is called *varimax rotation* where the rotation matrices come from the Lie group SO(2), for example:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{pmatrix} \tag{7.13}$$

Here, α is the rotation angle for the coordinate system. The rotation may be performed iteratively until an optimum is reached.

PCA is one type of factor analysis. Many variations do exist. Loadings may be extracted differently, represented differently and/or rotated differently. Furthermore, instead of using all eigenvectors as factors, only a subset may be used. The most frequently used *cutoff criterion* is the *Kaiser criterion*. It is based on the idea that in expressing the length of an eigenvector, the eigenvalue is a measure for the variance along the dimension of this eigenvector. In the initial normalization, variance was standardized to '1' for all variables of feature space. Hence, it makes sense to include only those factors in the loadings matrix that have an eigenvalue (variance) $\lambda_i > 1$. All other factors are actually more redundant (less variant) than the original data. Using a cutoff criterion higher than one causes further data reduction (and loss).

Above, we mentioned that factor analysis can also be applied on non-linear data. The algorithm so far is exclusively based on linear algebra. However, it can easily be extended to non-linear data by the introduction of a so-called *kernel function* $k(x, y)$. Kernel functions are explained in Chapter 18. They provide linear similarity measurement in non-linear spaces. PCA can be applied to non-linear data by setting $\chi = \frac{k(F',F)}{m-1}$ and performing the rest of the derivation as shown above.

The most important result of PCA for media understanding is the loadings matrix. This matrix can be employed to transform feature space to a more compact 'factorized' form by solving the matrix equation $W = F.X'$. Since the description elements of W are then not interpretable anymore (as being linear combinations of abstract codebook vectors), the transformed feature space is sometimes called *latent semantic*.[2] Hardly any redundancy (in terms of covariance) is left in the sparse representations generated by PCA.

The eigenvalues can be used as a quality measure for the feature transformations that generated feature space. The accumulated value $\sum \lambda_i$ is called the *communality*. The communality says how many per cent of variance the extracted factors explain. The fewer factors provide high communality (e.g. 90 per cent or more), the more redundant feature space is and the less effective the used feature transformations are – from the information-theoretic perspective.

[2] According to one colleague, this naming is a result of the reasoning that if multimedia scientists do not understand something, it must be semantic.

Practically, PCA is almost every able to reduce feature space to one third (if, for example, excellent audio transformations are used) or even one tenth (not untypical, for example, in the visual domain).

Factor analysis, in particular, PCA is a highly popular algorithm in media understanding. It is employed on audiovisual data as well as all other data types. Its information filtering effect is positive concerning dimensionality and performance. Noise is being eliminated effectively. However, throwing away data always bears the risk of increasing the semantic gap further.

In Chapter 16 we will encounter further methods for filtering of descriptions. The last section of this chapter, though, is dedicated to the understanding of what feature spaces typically look alike. This understanding is of fundamental importance for comprehending the difficulties of categorization.

7.4 Understanding Descriptions

In this section, we deal with two approaches to understanding the structure of descriptions: visualization and statistical testing. Visualization would certainly be the most intuitive way of getting into the data if there was not the problem that multimedia descriptions are usually high-dimensional and, since the maximum of visualizable dimensions is three, only fractions and views of the entire space can be visualized in one diagram. Statistical testing, on the other hand, is sensitive to the underlying model which may be conservative in the sense that the acceptance of a hypothesis may depend more or less on the size of the sample (here, a database of media objects). Still, tests for mean and variance may provide valuable insights into the structure of descriptions.

The motivation of understanding the descriptions is strongly connected to the categorization process. In the subsequent chapters, we will see that the shelf of categorization methods is well filled. Choosing the appropriate categorization method is of highest significance for the performance of media understanding systems. This choice depends on the structure of the data points: Is the space uniformly distributed? Where do accumulation points exist? Etc. The methods presented below help to answer these questions and, therefore, support the categorization process.

What is important to know about feature space? Certainly, as we already noted, the types of distribution of the description elements. There are two general options: *uniform distribution*, which is usually the desired one, and *normal distribution*, which should not be desired. The real distribution of description elements will, of course, hardly ever match one of these ideals. As part of the distribution, important properties are holes, i.e. values/intervals that are hardly or never used. Such holes can be made subject to compression or sparse representation, but they may as well provide valuable information on the quality of

the media database assembled in feature space. Eventually, if the objects in the media database are pre-labeled into categories, the distribution can be checked for each class of objects.

The *information visualization* research area provides a vast number of effective visualization methods, some of which even made their way into standard office programs. Professional statistics packages provide more-sophisticated methods and some, like the statistics language *R* were even developed for the visualization task. In media understanding we are primarily interested in the visualization of some feature space. Each media object should be visualized as a point in the high-dimensional space of description elements. The problem of mapping the high-dimensional space on two- or three-dimensional graphs can be solved in two fundamental ways:

- Show subspaces of one, two or three dimensions – if necessary, in sets of graphs. If only one dimension of feature space is shown than like a histogram against the set of values that occur on this dimension.

- Compress the n-dimensional description space to a two- or three-dimensional view. The major goal of this approach is to preserve the original distance relationships of feature space as good as possible.

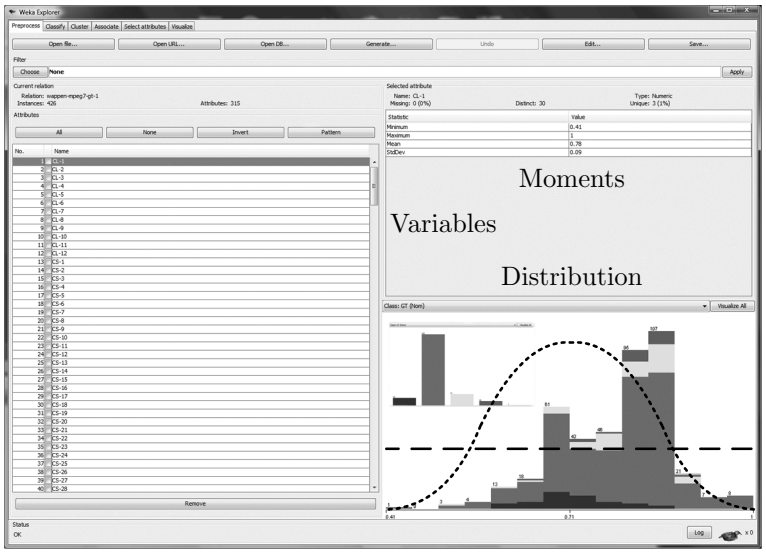


Figure 7.8: Data Visualization in Weka.

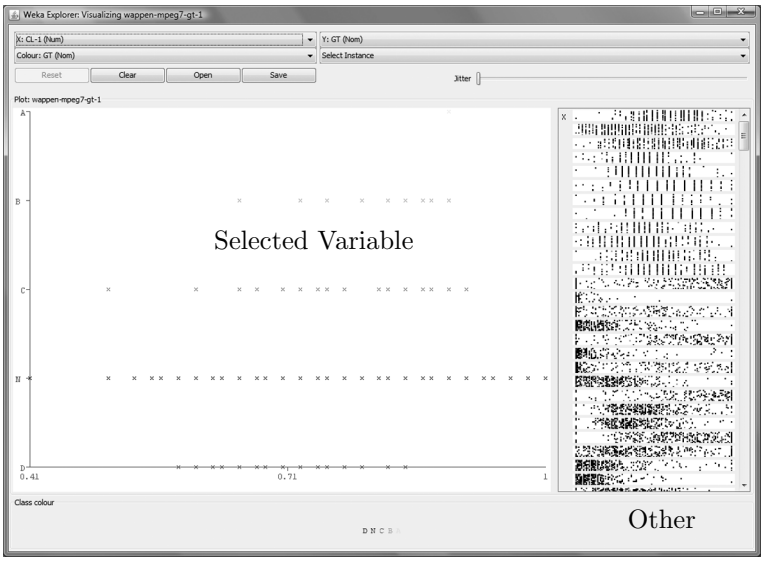


Figure 7.9: Data Visualization in Weka.

Below, we introduce one example for each approach. The *Weka Data Miner* [378] is a tool for general-purpose categorization. It provides a set of state-of-the-art categorization methods but as well some very nice visualization methods for feature spaces. Figures 7.8 and 7.9 show examples from a small visual media space. In the left panel of Figure 7.8 the description elements are listed by name (here, *CL* for MPEG-7 Color Layout and *CS* for Color Structure). The upper right panel shows the statistical moments of the selected variable (mean, span, standard deviation). The lower right panel provides a one-dimensional visualization. In the example data set, the media objects are labeled as members of five categories (coded in different colors). The bar segments in different colors express the histograms (distributions) of the classes and the entire population over the values occurring for the selected description element. As we can see, the distribution resembles rather the normal distribution than the uniform distribution. That is, most media objects have the same description value. Hence, this description element is not very discriminative between categories. Weka allows to browse through these individual visualizations of description elements quickly and, thus, to comprehend the topology of feature space in an intuitive way. It is, therefore, highly advisable to use this tool in media understanding.

Figure 7.9 shows a second view. Again, classes are coded in shades of gray. Pixels represent media objects. The right panel allows to grab the distributions

of twenty or more description elements with one look.[3] The left panel allows for mapping two description elements against each other. If the result is a diagonal line, the two elements are highly correlated. In the figure, we map one description elements against the class labels which shows nicely the distribution of the individual categories.

Multi-dimensional scaling (MDS) is one very effective approach for the second – holistic – approach to feature space visualization. MDS aims at compressing the description elements of media objects $f_i \in F$ into – usually two-dimensional – vectors x_i by solving the following optimization problem:

$$\min_{x_i} \sum_{i<j} (x_i \otimes x_j - f_i \bar{\otimes} f_j)^2 \tag{7.14}$$

That is, find vectors x_i that minimize the difference between the original distance relationships of the f_i. The quality of the mapping is operationalized as the squared distance. The expert reader will find it non-surprising that MDS was developed by specialists for human similarity perception. Applying the dot product (positive convolution) on the target vectors but L_1 distance (negative convolution) on the original vectors implements a so-called *dual process model* – which should according to the latest theory come closest to the human perception of similarity. See Chapter 28 for details. Hence, the down-mapping should preserve as much as possible of the way that humans would find characteristic about the high-dimensional feature space.

The remainder of this section is dedicated to understanding descriptions by statistical testing. We do not intend to provide a short guide to testing – such tutorials are available in masses on the Web. Instead, we would like to point out the most important tests, situations where they may be applied beneficially and the most relevant distributions that description elements may have.

We already mentioned the two major distributions that description elements may have. theoretically if the media objects under consideration are representative of their domain, the distribution of each description element should be *uniform*, i.e. each value should appear with the same likelihood. Then, feature space would be used ideally by distributing the categories over the entire space and maximizing the space between categories (theoretically). In practice, most description elements follow the normal distribution, i.e. some values appear with very high frequency and others with low frequency. The normal distribution has some properties that make it very interesting. Most importantly it has the highest entropy that a unit distribution may have. We will discuss this property and its meaning for media understanding in Chapter 22.

Two statistical tests are of outstanding relevance for media understanding: *t-test* and χ^2*-test*. The first tests if a set of numbers has a particular mean.

[3] Which would only be true if we were flies, as mentioned in Chapter 5.

The test value is compared against a Student distribution (similar to normal distribution), hence the conservatism of the test can be nicely controlled through the level of significance. The second test decides if a set of numbers has a particular distribution. Here, the test value is an aggregated sum of squared errors of the actual distribution against the given distribution. If the test value exceeds the value of the χ^2 distribution with $n-1$ degrees of freedom (n being the number of variables), the hypothesis is refused.

Checking for a particular mean is important for identifying description elements that actually measure the same property. Only one such element needs to be included in more advanced filtering techniques such as factor analysis. Checking for a particular distribution is useful, if the feature transformations under investigation produce nicely distributed descriptions. However, in media understanding practice this is seldom the case. Hence, it makes more sense to apply a multivariate method such as *analysis of variance* (ANOVA) which provides statistics on the pair-wise dependencies between description elements in terms of means and standard deviations. It is highly advisable to investigate feature space by ANOVA prior to categorization.

Powerful methods exist for information filtering of descriptions. The conclusion of this chapter is, therefore, to apply all feature transformations on the given media data and extract as long descriptions as possible. In the information filtering process these descriptions can efficiently be reduced while preserving most of the original information. Furthermore, visualization and statistical testing allow for understanding the characteristics of the feature space, which is important for selecting the best categorization method for the data. Many categorization methods do exist. The general concept and the most basic algorithms are presented in the next chapter.

Chapter 8

Simple Categorization Methods

After the introduction of some fundamental concepts of categorization we discuss methods based on decision rules, on vector spaces and distance measurement and on dynamic association of description elements.

8.1 The Setting of Categorization

At this point of the media understanding process we have extracted descriptions of signals by feature transformations. The signals are media objects drawn from a media database. The descriptions may be composed of quantitative (interval-scaled) and/or symbolic (nominal-scaled) elements. We have then merged the individual descriptions and removed the redundancy by information filtering. Now, the last step required to complete the big picture of media understanding is *categorizing* the descriptions of the media database into *classes* by a *classifier*. The resulting *class labels* express a semantic category of the source media objects.

This chapter and the next chapter introduce a particular selection of algorithms and models for categorization. This chapter's focus is on simple models – most of them are very old. In terms of machine learning that means up to 60 years. Since they are easy to comprehend, they are ideal for being used in an introductory chapter. The next chapter, however, introduces probabilistic categorization methods that are not at all simple (or outdated). Some of them are indeed the state-of-the-art in some fields of media understanding. For example, hidden Markov models are still leading in speech recognition. Our decision to

put the chapter on probabilistic categorization into the first part is based on two facts. Firstly, the rules behind probabilistic methods are few, and as soon as they are well understood the entire domain can easily be understood. Secondly, probabilistic methods can be applied very effectively. The training may be tedious but as soon as a categorization model is available, it can be employed very efficiently. Therefore, probabilistic methods are in terms of their application related to the simple methods discussed in this chapter.

The goal of categorization is clear. It is the reduction of a high-dimensional description vector to a single number, the class of the media object. Here, class connotates the existence of some *context* – above, we named it *semantic meaning*. That is, the class label is only meaningful in relation to the question of the particular media understanding problem. For example, if the problem is face recognition, some class label 23712 may refer to a person with the name *John Drake*. If the problem is sentence analysis, some label 2 may stand for *Question*. We can conclude that the categorization process is a *specialization process*, where data are transformed into particular meanings. Both categorization method and class label are only meaningful in the direction of the specialization and the number of possible specializations is as large as the number of reasonable media understanding problems. Each media understanding process requires its particular categorization process. The fundamental approaches discussed here, in the next chapter and in the other parts of this textbook have to be parameterized and trained into the direction of the specialization. That is, categorization methods – in contrast to feature extraction methods – must be based on highly parameterizable models. Why not feature transformations? The purpose of feature transformation is the efficient representation of media content. It is not yet the task of feature transformation to perform the adaptation of the media data into the direction of the media understanding problem.

Still, categorization and feature transformation are related in the sense that both steps perform information filtering. In the mathematical notation, we stated that media objects, descriptions and class labels are closely related. Since this *data view* on the media understanding problem is inverse to the *process view* the processes must necessarily also be related. In conclusion, the major technical difference between feature transformation and categorization is that the latter process allows more flexible specialization.

It is important to note that *whatever the media is*, the same categorization techniques can be used. Text retrieval, for example, employs models similar to those used in face recognition, P300 detection and music genre classification. Hence, we do not differentiate categorization methods by their input data. This statement is not necessarily true for descriptions. If the input media database is described by a static feature space of corresponding elements, almost any categorization method can be applied. If the length of descriptions depends on the media source, not all categorization methods are applicable. Then, three

approaches are available:

1. Add white spaces (zeros) where necessary to the description until all descriptions are of equal length.
2. Define categories (e.g. words) and aggregate a histogram.
3. Apply a dynamic categorization method like those discussed in the last section of this chapter.

The first strategy is not always possible. For example, in text retrieval it is not clear how missing adjectives should be treated. In this case – mostly, for symbolic media – another solution has to be chosen. See Chapter 16 for more details on this issue.

The world of categorization can be approached from many different starting points. One is to browse through the population of methods and categorize them by their principal approaches. Following this path we come up with four principal approaches:

- Rule-based differentiation
- Similarity-based grouping
- Probability-based grouping
- Neural discrimination

The last type of approaches will be discussed in Chapters 26 and 29. Probability-based approaches are the topic of the next chapter. The second group – by far the largest in terms of different models – is discussed below and in Chapters 18, 19. Rule-based approaches are discussed in Section 8.2.

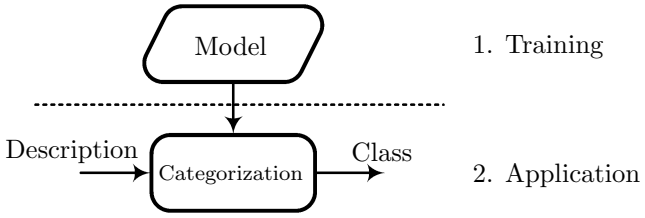


Figure 8.1: The Categorization Process.

Figure 8.1 summarizes the setting of categorization. We focus on the most important aspects only. A more thorough discussion will follow in Chapter 17. The actual categorization process transforms a description into a class. In order

to fulfil its job, the categorization process requires a model. The model has to be built in a training step – in most cases. Some very simple categorization methods do not require training. In consequence, these methods are not very flexible when it comes to *specialization* to a particular context.

The sequence *training, application* is significant for most machine learning approaches. The training step fulfils what we discussed above: the parameterization of a general model to a particular semantic meaning. In order to perform the training, often, *additional data* are required. There are two major sources of training data:

- References
- Ground truth (GT)

References are description vectors with the same morphology as the description vectors of the feature space. One or a group of references refers to one particular class. That is, references link classes to *prototypes* of media events (e.g. an average face). Reference vectors may be drawn from feature space or be artificial vectors (defined by hand or randomly).

Ground truth is basically a set of reference vectors, each with its class label attached. Often, more than one vector is given per class. GT is usually assembled by human beings that rate particular media objects as members of some group. The GT data set is built by extracting the descriptions of these media objects and adding the class labels as extra elements. From the philosophical perspective, ground truth is an awkward denomination, since however big the set and no matter how many humans are involved in the labeling of the media objects, no data set will ever come close to the truth behind any media understanding problem. This pessimism is based on the fact that human judgments of stimuli are subjective and to a large degree vulnerable to arbitrariness. We can certainly not assume that the GT provider is a rational decision maker.

However, practically GT is the best we can have for the training of classifiers. In the optimal case, it was extracted by quantitative methods from a sufficiently large group of representative people. Very often, the GT for some new solution to a media understanding problem is based just on the scientists who developed the method. Such a foundation is, of course, not satisfactory. We cannot, though, put the blame on the scientists alone. Today, it is impossible to gain merit for assembling a high-quality ground truth for some domain. Such an undertaking is not attractive for researchers. It is usually not paid for and many reviewers do not ask for it anyway. Still, serious media understanding researcher should always make sure that the ground truth employed for training and evaluation is of the highest quality possible.

Whatever we have, references or ground truth, is for classifier training split into two parts. The *training set* is employed for specialization of the catego-

rization method. Then, the typically larger *test set* is employed for estimating the quality of the trained classifier. If the quality is unsatisfactory, the entire process is repeated with altered parameters or a different categorization method. Evaluation procedures are discussed in Chapter 10.

Another important aspect of the setting is the fact that the entire process has typically two levels:

- The *macro level* of discrimination of entire media databases
- The *micro level* of individual assessment of descriptions

The micro level is embedded in the macro level. In the process of discrimination of the entire data set under consideration the micro process is executed for individuals of the population. Practically, the micro level may be the judgment of individual descriptions, the comparison of one description to a reference or the pair-wise comparison of descriptions. The important fact is that the micro process is more or less independent of the macro process. Hence, the same micro process, for example, distance measurement, can be found in a number of different macro processes. We differentiate the last two sections of this chapter – both dedicated to similarity-based categorization – into methods that lay more weight on the micro process and methods characterized by their macro process.

Eventually, the macro process is responsible for the flexibility of a categorization method. Specialization is doubtless required and desired by a good classifier, but limits do exist. Flexibility is usually measured on a scale with the extremes *rigidity* and *overfitting*. A rigid macro process is not able to adapt to the requirements of a particular media understanding problem. The results are always the same independent of any training. In consequence, no references nor ground truth is required. One example for a perfectly rigid macro process is *cluster analysis* (see below). On the other end, a perfectly flexible process is prone to overfitting, i.e. too close adaptation to the ground truth. Then, every new instance of a media event is categorized exactly as the ground truth data. If the GT was perfect, this behavior would be desirable. Due to the practical limitations discussed above this is unfortunately almost never the case. What is practically desired, is a classifier that *generalizes* well. Such a classifier is neither rigid nor overfitting. It is flexibly adaptable to the constraints of a particular domain but not completely dependent on the training data. Machine learning research is still searching for this classifier.

Are there any particulars for symbolic descriptions? Not generally. On the macro level, if the symbolic descriptions are not of fixed length, it makes sense to normalize them by histogram aggregation or by introducing white-spaces where necessary. The more important adaptations have to be made on the micro level. Rule-based approaches can deal with any type of data. Similarity-based,

probability-based and neural process, however, may require some type of *quantization* of the symbolic data. In similarity-based categorization, a number of successful symbolic measures have been defined. These will be discussed in the last two sections of this chapter. Probability-based methods require the first-time transformation of symbols to *likelihoods of occurrence*. Again, numerous solutions have been developed over the last decades. Eventually, for neural application symbols need to be transformed to a binary representation which can be performed easily. The macro process of categorization is not affected by these means of quantization.

In conclusion, categorization methods make descriptions specific to some context expressed in training data such as ground truth. Classifiers have to provide models flexible enough for this adaptation but rigid enough for avoiding overfitting. In the next section, we introduce rule-based solutions for this problem.

8.2 Rule-Based Categorization

Rule-based categorization methods differentiate classes by conditions of the following form:

```
if f < ϵ then
 follow left branch
else
 follow right branch
endif
```

Here, f is a description element, ϵ is a threshold, the left and right branches may be conditions, trees of conditions or class labels. These conditions are also called *weak classifiers* or *decision stumps*. We will investigate them in greater detail in Chapter 19. The result of nesting such conditions is a *decision tree*.

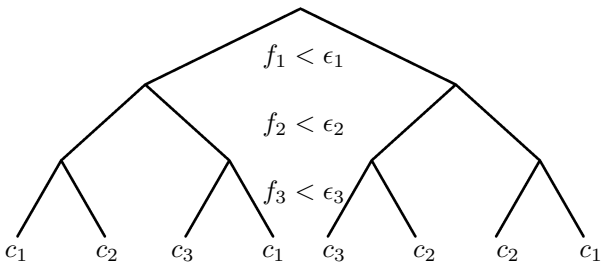


Figure 8.2: Decision Tree Example.

Figure 8.2 shows an example of three layers. As we can see, different description elements are investigated at different levels, and multiple paths may lead to the same class label.

Decision trees can easily be constructed from ground truth data by a bottom-up strategy. That is, all descriptions in the ground truth with the same class label are grouped. Then, for each class, a condition is defined for the most invariant element of all group members. The tree is constructed iteratively by joining conditions on the same description element or merging independent subtrees by choosing a new description element for the next level. Such a strategy can be implemented easily and quickly. Typically, decision trees have much fewer levels/dimensions than the description vectors. That is another reason for the high efficiency of decision trees.

Obviously, the decision tree is a very powerful method. The micro level is just comparison of description elements to a threshold. The threshold is set during the training process, static, and different from level to level. The macro level is a sequence of comparison operations. This scheme is as well applicable on quantitative data as it is on symbolic data and on descriptions of variable length. The application can be performed as quickly as the training process.

Against these advantages stands one major drawback. Decision trees are very prone to overfitting. The quality of the categorization outcome depends completely on the quality of the ground truth. Since in practice almost any ground truth is only partially true, the categorization is also only partially correct. Decision trees do not provide any form of rigidity. Their generalization behavior is therefore very bad.

Various approaches exist to overcome the overfitting problem of decision trees. The *random forest algorithm* is one very popular approach. It consists of the following steps:

1. Select n subsets f_i of the ground truth.

2. Train a decision tree c_i for each set f_i.

3. Categorize descriptions with the most frequent label produced by all decision trees c_i.

The random forest algorithm is a composition of small decision trees. In machine learning, such an approach is called an *ensemble method* (see Chapter 19 for details). Since it applies a divide and conquer strategy during training its performance is worse the performance of a decision trees. However, the usage of an ensemble of *specialized* decision functions introduces some degree of generality. Each decision tree expresses a particular point of view. The different natures of the ground truth segments may cause variable views in the random

forest. As a result, the joint decision may be more general than the decision of a global tree.

Decision trees and random forests are used in many media understanding applications today. The training is usually performed on a rather small fraction of the ground truth while the rest of feature space is used as the test set. Decision trees are particularly effective for minor media understanding problems (such as classifying photos of two types of vegetables). They are, therefore, highly recommendable for student projects. However, for big problems such as unconstrained video surveillance decision trees can at most serve as the macro process in a cyclic media understanding process. For example, based on some other categorization, a decision tree could decide whether an observed event is dangerous or not.

In conclusion, the biggest advantages of decision trees are excellent results for limited ground truth and efficiency, in particular, on high-dimensional data sets. These advantages are paid with bad generalization, i.e. a remarkable tendency to just representing the ground truth. In the next section, we will introduce methods that are close to the other extreme of categorization, perfect rigidity.

8.3 Distance-Based Categorization

This section summarizes methods where the micro process is distance measurement of corresponding description elements. The macro process is variable and depends on the type and number of available training data. The methods gathered in the next section do also employ distance measurement but in a dynamic form. Not corresponding elements are compared but elements with optimal distance (depending on the optimization goal).

This section starts with a brief introduction of distance measures for similarity measurement. We describe cluster analysis as an example for a categorization method applicable without training data or test data. Then, the vector space model, k-means and k-nearest neighbor categorization are discussed for situations where exactly one reference, n references or ground truth is available for training and application. More complex categorization methods of this type are introduced in Chapters 18 and – partially – 19.

The two central ideas of all distance-based methods for categorization are that distance is a measure for similarity and the distances between objects and/or references can be employed to group (cluster, classify) them.

There is a lot that could be said about the relationship of similarity and distance. It has been subject to psychological research for more than eighty years now. Some of the details will be discussed in Chapter 28. The most important fact is that, for humans, distance m^{-1} is certainly not inverse similarity m. The relationship is rather seen as exponential: $m = e^{f(m^{-1})}$ where f is a negative

function. These findings have been neglected in distance-based categorization so far with the effect that machine categorization produces – unsatisfactory – results significantly different from human expectations.

Distance measurement is usually based on *metric functions*. Such functions have been designed in mathematical research since the mid-nineteenth century. Therefore, a large number of variations is available today. See the first section of Appendix B for a collection of metric (and non-metric) distance functions. These measures are systematically discussed in Chapter 28.

A measure m is called a *metric* if it fulfils the following conditions:

$$m(x,x) = m(y,y) \tag{8.1}$$
$$m(x,y) \geq m(x,x) \tag{8.2}$$
$$m(x,y) = m(y,x) \tag{8.3}$$
$$m(x,z) \leq m(x,y) + m(y,z) \tag{8.4}$$

These conditions (variations in formulation are possible) are called the *metric axioms*. They are perfectly reasonable if distance measurement is seen as a rational approach for measuring the length of the path between two points x, y. However, if distance measurement is employed for modeling similarity (a non-rational, psychological concept), the metric axioms are too rigid. For example, it has been proven wrong that symmetry (the third conditions) holds for all stimuli. It is rather the case that more complex stimuli are always found less similar compared to others than the other way around. See Chapter 28 for details. Still, mostly metric distances are employed for distance-based categorization today.

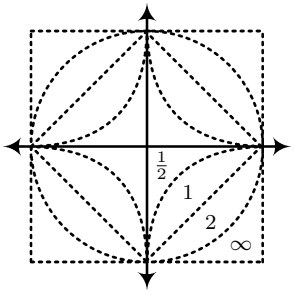


Figure 8.3: The Minkowski Distances.

One particularly important group of metric distance measures is the Minkowski distances group. The family of parameters a_1, a_2 is defined as follows (measure Q1 in the appendix).

$$\sqrt[a_2]{\frac{\sum_i |x_i - y_i|^{a_1}}{K}} \tag{8.5}$$

Traditionally, $a_1 = a_2$. Psychologists have found out that sometimes a root different from the exponent describes the human perception of distance/similarity better. Figure 8.3 illustrates the four most important Minkowski distances for $a_1 = a_2$. If $a_{1,2} = 1$ we receive the city block distance, if $a_{1,2} = 2$ Euclidean distance, and if $a_{1,2} = \infty$ the Chebyshev distance. If $a_{1,2} < 1$ the Minkowski distances become non-metric. This case was not intended originally but, again, psychologists could show that under some circumstances distances with an exponent smaller than one fit better to human distance perception.

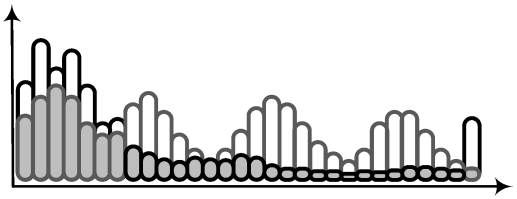


Figure 8.4: Histogram Intersection Example.

Table B.1 in the appendix lists all major distance functions. One that is of particular interest to us is *histogram intersection* (Q15): $\sum_i \min(x_i, y_i)$. This method simply counts the overlapping regions of two data sets. Figure 8.4 illustrates the approach where the gray area is the overlapping part. Today, this method is frequently used for comparing color histograms in visual media understanding. It can be applied to any histogram data (e.g. correlograms, edge histograms) and usually produces good results, i.e. a categorization that appears reasonable to humans.

In the remainder of this section, we introduce four methods for categorization that employ distance functions. The first method requires no training data. The next two require references, and the last one requires a ground truth.

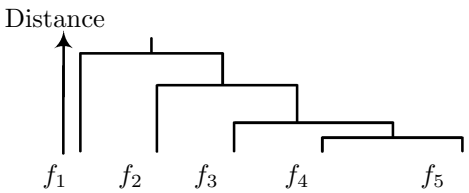


Figure 8.5: Dendrogram Example.

The first categorization method to be discussed is *cluster analysis*. Figure 8.5 gives an example for the result of a cluster analysis: a *dendrogram*. The five media objects f_i are related by their distance. The vertical position of the horizontal line connecting two elements or branches denotes the distance between the two objects. For example, the distance between f_4 and f_5 is small, while the distance between f_1 and f_5 is very big. The morphology of the illustrated dendrogram is typical for the method. Two objects are found most similar, and all others are set in relation to them.

Cluster analysis requires no training data. In fact, the method leaves no space for knowledge other than the descriptions. It is, therefore, a very practical starting point for categorization. It is always advisable to run a cluster analysis on a newly created feature space, because cluster analysis reveals the overall structure of the data. If the majority of the objects in feature space are closely related (like f_4,f_5) then the chosen feature transformations are obviously not discriminative enough. Ideally, the dendrogram of a feature space should consist of several near equally-sized group on levels of average distance (the classes). If no classes can be identified in the dendrogram, then they probably do not exist, and the employed feature transformations are not suitable for distance-based categorization.

Several algorithms do exist for cluster analysis that are based on two construction principles: *top-down* (*separative* clustering) and *bottom-up* (*agglomerative* clustering). In the first case, the dendrogram is built from the top. In the first step, the element with the largest distance is separated from the others, in the second step the next, and so on. The algorithm can also be run recursively by separating groups of n objects and running the algorithm again for each group. In agglomerative cluster analysis first, each element of feature space is considered a class of its own. In the first step, the two nearest classes are merged and so on. Mixed forms of bottom-up and top-down clustering exist as well.

Cluster analysis is a straightforward procedure for the categorization of feature spaces. In fact, the resulting dendrogram does not define classes yet. These have to be defined by hand by setting thresholds of maximum distance between branches of the dendrogram. The method has no parameters, is simple to use and very recommendable as a starting point for understanding the structure of a feature space.

The second distance-based categorization model is the *vector space model* (VSM). The VSM was defined in the 1960ies for text retrieval. The idea is very simple. In a database of media objects, we search for a *query example* and return a *result set* of the n most similar objects. The operationalization is as follows: The query example is one reference description vector with exactly the same structure as the descriptions of the media objects in the database. Only two classes are categorized: *relevant* are the members of the result set, *not relevant*

is the rest. The *retrieved* objects are interpreted as the relevant objects. In Chapter 10 we will see if this assumption is realistic.

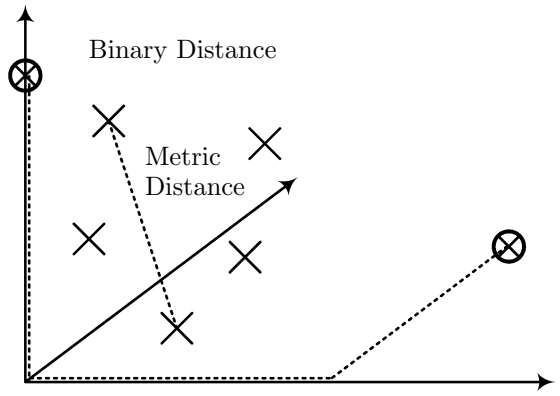


Figure 8.6: Vector Space Model Example.

Figure 8.6 illustrates the VSM categorization process. Descriptions are interpreted as vectors in a (non-)metric vector space. That is, the dimensions are assumed to be at least interval-scaled. Therefore, distance and dot product can be defined. Similarity is measured as the inverse of distance. Typically used distance measures are city block distance and Euclidean distance. In the figure, the direct dotted line between two objects (denoted as x) expresses their distance. If one of them is given the role of query example, then the result set is the circle (or ellipse, depending on the type of distance measure) around it that covers exactly n other objects. Value n is the only parameter that needs to be set. In the VSM, the most similar object is always the query example.

One special application of the VSM worth mentioning here is *binary retrieval*. In binary retrieval, all dimensions of the vector space are just predicates of the form $\{0, 1\}$. That is, each property (dimension) either exists or not for a particular object. Binary retrieval is tailor-made for text understanding. Each possibly occurring term is modeled as a binary description element. For a modern language, the resulting description of a text segment may have several hundreds of thousands dimensions. The rest of the process is performed as in the normal VSM. See the line connecting the two circles in the figure for an example. The result is, of course, a similarity measurement process that employs one of the predicate-based measures listed in Appendix B.2. We will discuss these measures in Chapter 28 in detail. For the moment, one very popular measure is the Hamming distance which is equivalent to the city block metric in the continuous domain.

In summary, the VSM extends cluster analysis by one reference object and

one parameter. The result is a categorization model for the differentiation of relevant (with respect to the query example, few objects) and irrelevant objects (many objects), which is the typical scenario of *retrieval*.

The next step in complexity is the *k-means algorithm*. This categorization method requires one reference vector per class, i.e. in the simplest case two vectors. The fundamental idea is very simple. Each description in the feature space is compared to each reference vector. The reference vector with the lowest distance wins and the media object represented by the description vector is added to the class represented by the reference vector.

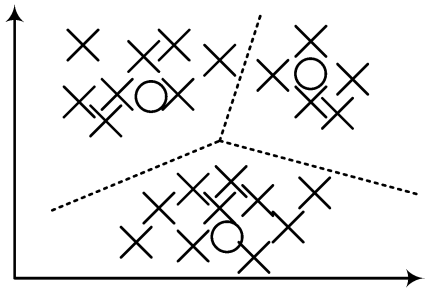


Figure 8.7: K-Means Categorization Example.

Figure 8.7 illustrates a k-means example. The circles are the reference vectors. The crosses are the members of feature space. The example employs Euclidean distance. Since we have three references, the results are three classes of varying size. K-means categorization is not fuzzy. Each description vector is attached to one reference. In the unlikely case of exactly the same distance to two or more reference vectors, some second-order decision taking process has to be used. The result of k-means categorization is called a *Voronoi tessellation* – in the figure indicated by the dotted lines.

The k-means algorithm is a simple extension of the VSM, yet it can be very powerful. The performance of k-means depends exclusively on the wisdom of choice of the references. Sometimes, the reference vectors are distributed uniformly over the space, sometimes initialized randomly and sometimes by a cluster analysis process. Generally, it would be desirable that inaccurately positioned references can be moved during or before the application of k-means. An extension that implements that is the *self-organizing map* discussed in Chapter 19.

In summary, k-means is a simple, quick categorization algorithm for $n \geq 2$ classes that requires no training and may deliver excellent results if the reference vectors are chosen appropriately. Of course, the algorithm can easily be extended to representing classes by more than one reference – for example, by using the

mean vector of all references associated with a class. Alternatively, the references can be applied separately but associated with the same class labels.

The last distance-based algorithm to be discussed here is the *k-nearest neighbors* algorithm (k-NN). K-NN is related to k-means but requires more-sophisticated training data. Instead of a few reference vectors, we require a ground truth, i.e. a feature space in which each object has a class label. On this foundation, the k-NN applies a simple association algorithm. Each new description is positioned in the underlying vector space and then of the k nearest neighbors in the ground truth (i.e. the would-be result set of VSM, if the new description was the query example) a histogram of the class labels is computed. The class with the largest number of neighbors wins. The media object connected to the new description is labeled with the winning class.

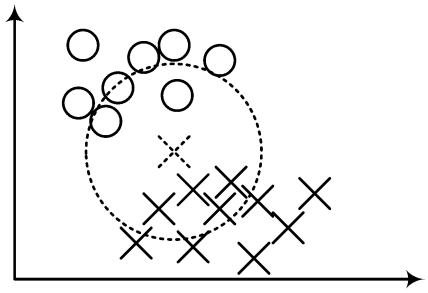


Figure 8.8: K-Nearest Neighbor Categorization Example.

Figure 8.8 gives an example of k-NN. The new vector is dotted. The two groups are given as circles and crosses. Since more crosses can be found in the neighborhood $k = 7$ the new vector is considered a cross. The advantages of k-NN are obvious. The algorithm is very simple, very fast and requires no training. On the other hand, it makes relatively little of the complex ground truth. Other methods, such as those discussed in Chapter 18 employ sophisticated training in order to optimize the quality of the categorization process.

Cluster analysis, VSM, k-means and k-NN are four highly related algorithms. All four are based on the vector space assumption, employ distance measures and rely on the constraint that corresponding description elements have the same meaning. The major difference is the training set. While cluster analysis needs nothing and VSM, k-means can do with a few references, k-NN requires an entire ground truth. In return, cluster analysis gives only a general overview over feature space. VSM is tailor-made for retrieval applications. K-means and k-NN can be very effective (and practically relevant) if the references/ground truth data are selected with care. In the subsequent chapters, we will encounter a number of methods that are based on these principles.

In conclusion, we presented a number of categorization methods that are based on distance measurement. The common advantage of these algorithms is their good performance due to the simple models employed. Furthermore, they scale relatively well with increasing dimensionality of feature space. Their common drawback is their simplicity, which may increase the gap between media semantics and descriptions even further in the categorization step.

In the next section, we will likewise encounter categorization algorithms that employ distance measures. However, there, some optimization criterion is added to the measurement process that leads to a selection problem of the form: Which elements of two descriptions fit together best?

8.4 Dynamic Association Models

Rule-based categorization is applicable on any type of data. Distance-based categorization requires the definition of some measure m. If m can be defined reasonably, then the entire zoo of methods can be applied no matter if we deal with text, bioinformation, visual descriptions or some biosignal. That is equally true for the dynamic association models discussed below though these models provide one degree of freedom more than the static distance-based models. The attribute *dynamic* stands for dynamic association of best-fitting elements of two descriptions. That is, the micro process of categorization is embedded in an *optimization process* that aims at identifying the best mapping between two *sets of elements* – here, not arrays!

Normally, *dynamic* connotates change over time. That is not the case for the models discussed in this section. Categorization methods that change (learn) over time are discussed in Chapter 19. The methods of this section are only dynamic in terms of association.

As we will see below, the micro processes and the optimization processes of dynamic association models are usually well defined. The macro process, however, may only exist rudimentary. For this reason, dynamic association methods can also be seen as plug-ins for the micro process of some other method. For example, the bag of words methods can easily be embedded in a k-nearest neighbor algorithm. Many other combinations are thinkable.

In the remainder of this section, we deal with three types of dynamic association models. The first type is the *bag of something* method of which we have already seen an example in Section 5.1. The second type is the *dynamic warping* method, which is very popular in bioinformation processing as well as audio retrieval. The last type is the group of *similarity meta-models*, which are, for example, employed in shape recognition.

The bag of something methods, in particular *bag of words* if the descriptions consist of text or *bag of features* if the descriptions are quantitative, are

somewhere in the middle between dynamic association models and traditional distance-based methods. They certainly employ an optimization criterion, since for every part of the description of one object the best match is identified in the other object. The best match may be expressed as the lowest distance or best sub-string match, for example. On the other hand, the optimization criterion need not be (and normally, is not) applied on the element level (one symbol, one quantity). Rather, groups of elements (so-called concepts) are formed (e.g. words, phrases, color histograms of sub-images, etc.) and the optimization is performed on the group level. That is, if the entire media content is defined as one group, we arrive at normal distance measurement. If the groups are broken down to individual symbols, we arrive at a real dynamic association process.

The bag of something methods work for a pair of media objects and predefined concepts precisely as follows:

1. Identify the concepts of one object in the other object.
2. Summarize the number of matches. If necessary, matches can be grouped again (e.g. spatially, by quality, etc.).

The number of matches is a measure for the similarity between the two objects. Please note that the bag of something methods do not prescribe a particular distance measure for the micro process. Often, Minkowski distances are employed but any other measure can be used instead. Therefore, the bag of something methods are plug-ins of some macro process that may use any micro-process as plug-in. This very flexible scheme is heavily used in text understanding and on so-called local features which we will encounter in Chapter 14.

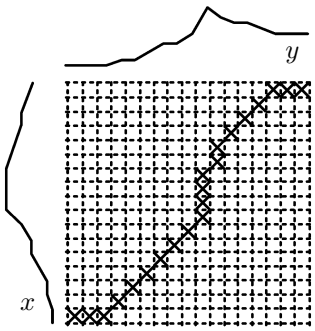


Figure 8.9: Dynamic Time Warping Example.

The dynamic warping methods are dynamic association models par excellence. We named this group after one approach very popular in audio classification: *dynamic time warping* (DTW). Figure 8.9 gives an example for the

algorithm in which two objects (represented by descriptions x, y) are compared pair-wise by their descriptions. The depicted data may stand for any type of description (either quantitative or symbolic). Now, the optimization goal of DTW is to identify the best match between the two description *sequences* while allowing *insertions* and *deletions*. That is, the algorithm is not totally ignorant of the vector structure of the descriptions.

The recursive DTW algorithm solves the following problem:

$$m_{dtw}(x, y) \rightarrow min \tag{8.6}$$

with

$$m_{dtw}(x, y) = m(x_i, y_j) + min \begin{cases} m_{dtw}(x_{i-1}, y_{j-1}) & match \\ m_{dtw}(x_{i-1}, y_j) & insert \\ m_{dtw}(x_i, y_{j-1}) & delete \end{cases} \tag{8.7}$$

Here, m is some measure and the counters i, j start at the last elements of x, y. By the way, x, y need not be of equal length. In the straightforward case of a diagonal walk we have a match. From the perspective of object y the second possibility for minimization is an insert of a copy of the actual symbol, since one more element of x is mapped on the current one of y. The last option is considered a delete because one element of y is ignored. The resulting m_{dtw} accumulates the optimal distance through the matrix of possibilities (illustrated in the figure).

The DTW problem is usually solved recursively. The currently best algorithm is only of little interest to media understanding. However, employing a recursive algorithm is not just natural to the problem definition, but it allows to reduce the magnitude of the problem from $O(n^2)$ to $O(n. \log n)$.

Dynamic time warping is used frequently in audio retrieval for matching the descriptions extracted from spoken words. Such descriptions are typically not of uniform length, which makes DTW one of the few options for efficient categorization. In particular, on resource-limited systems DTW is a good option for categorization, because then, the state-of-the-art Markov processes (next chapter) are often not applicable.

So far, we focused on DTW as one dynamic warping method. The abstracted model of DTW is the so-called *edit distance*. The general principle of the edit distance is that the difference between two objects is the result of editing one of them. Therefore, the similarity (distance) between a pair of objects can be measured by the number of operations required to transform one into the other. In DTW, we had the three operations *match, insert, delete*. A second model, the *Levenshtein metric* suggests the three operations *insert, delete, substitute*. This metric is usually employed on symbolic descriptions (primarily, text). It is considered a generalization of the Hamming distance (measure P3 in Appendix

B.2) which is used in typewriter programs for identifying the best match of incorrectly spelled words. Like DTW, the Levenshtein metric requires a dynamic programming algorithm for the identification of the best solution.

Edit distances are currently heavily investigated in psychological research. For example, the authors of [147] suggest to combine edit distances with the *human choice model* of the following form:

$$m_{edit}(x,y) = \frac{m_{edit}(t,y)}{m_{edit}(t,x) + m_{edit}(t,y)} \tag{8.8}$$

Here, t is some reference stimulus. This model allows to introduce some semantic context (expressed in t) into the measurement process. The authors claim that this model fits *human structural alignment*. In Chapter 28 we will hear more about choice models, structural alignment and their combination.

One reason why structural alignment and edit distances are now a hot topic of psychological research may be that the same dynamic warping approach is employed in bioinformation understanding for *sequence similarity* measurement. There, the goal is to measure the overall similarity of two gene strings (or substrings). The current state-of-the-art approach is the *Needleman-Wunsch algorithm* which is nothing else than dynamic time warping where the Levenshtein metric is used for distance measurement. The only particular aspect of this algorithm is a penalty score for insert and delete operations.

Furthermore, bioinformation processing employs a very similar algorithm for local alignment of gene strings. Local alignment aims at identifying the best matches between two sequences of symbols. The solution, the *Smith-Waterman algorithm* is again a DTW process with one additional condition. Empty substrings are aligned at zero costs.

In summary, dynamic warping is of high relevance for both quantitative and symbolic data. The optimization process aims at optimal structural alignment of a pair of descriptions. The micro process may employ any similarity function even though edit distances are frequently used on symbolic data. Dynamic warping can be used as a stand-alone categorization method but likewise, be embedded in any other process. The most frequent application domains are speech and bioinformation.

Figure 8.10 gives an example for a third type of dynamic association models: similarity meta-models. The figure illustrates the measurement results of two typical models. The Hausdorff distance and the bottleneck distance (listed as M5 and M6 in Appendix B) consider the two given descriptions as sets of elements (visually, points). They measure the typical distance of the two sets by choosing largest (supremum) and smallest (infinum) distances m_i between pairs of points. The distance is eventually represented by the relationship of just two points. The Hausdorff model is frequently applied in visual shape retrieval. Both measures may be applied for graph matching and related tasks. Which method to choose

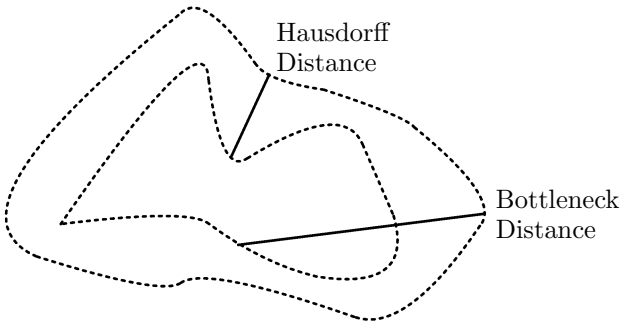


Figure 8.10: Shape Comparison by Distance Meta-Models.

depends on the desired goal. The optimal form of selection is the definition of a thorough ground truth, evaluation of all available categorization models and selection of the one that fits the ground truth best at acceptable overfitting.

Two further similarity meta-models are M2 and M7 in Appendix B.3). The first one, the Catell measure is not really a dynamic model. It is more a normalization of distance measures. However, the meta-model may be subject to optimization, which may alter the distance relationships expressed by m_i.

Measure M7 is the *Mallows distance*, which is (slightly differently defined) also known as the *Wasserstein distance* or the *earth mover's distance* (EMD). In particular, the EMD is of highest significance in audiovisual media understanding. It is employed for all kinds of histogram comparisons where it tries to identify the global minimum of a product of distance and cost of transfer. Distance is defined by a measure m_i. Cost of transfer (for example, between histogram entries) is defined by a function c. Over all permutations of one description vector, EMD searches for the minimum of distance to the second vector while taking the costs of permutation (movement by unit) by function c into account. The name *earth mover's distance* illustrates the principle well. The total cost of moving earth from one point to another is constrained by the distance between the points and the costs of transport.

The EMD shows similarities to a number of other models. First of all, it is related to histogram intersection. Both methods endeavor to identify the common aspects of two descriptions. Secondly, it is related to the edit distances. The cost function introduces a degree of freedom that may be interpreted as some sort of manipulation. Moving values from one element to another can be interpreted as a sequence of one delete operation and one insert operation. Therefore, the EMD (generally, the Mallows distance) may also be seen as a dynamic warping model.

One last similarity meta-model that we would like to mention in this section

was defined in [143]. The authors propose a similarity model that originates from text understanding but relies on a distance-based micro process. The optimization algorithm consists of two steps. In the first step, all objects in feature space are compared pair-wise. In the second step, the best matches are aggregated. Here, *best* is defined by a set of criteria partially motivated by insights on the nature of human similarity perception. The authors propose a particular micro process, a measure called *systematic similarity* that resembles the human choice model. However, any other model may be employed instead. This model appears interesting though for practical application some hurdles have to be taken. The descriptions have to be provided as a hierarchy of entities and attributes (due to the computational linguistics' origin of the method). Selection of the best description elements is very important for the performance of the measure. Furthermore, the measure depends on a threshold μ_0 which has to be set with care. On the other hand, the algorithm can be employed as some kind of edit distance as well, which makes it interesting as an integrative approach.

In summary, similarity meta-models can be employed to categorize any type of data. The most frequent domains are text understanding, visual shape retrieval (for example, by interest points, see Chapter 14). One question that arises is: Which method should be used when? This is, eventually, a question of experience. The experienced visual retrieval researcher will know when to employ the Hausdorff distance and when, for example, dynamic warping. For the beginner, one reasonable advice appears is to try all available methods and choose the one that performs best. We will develop the details of this scheme further in the Chapters 11 and 21. For practical application, Weka [378] provides a good starting point, since this package implements many important categorization methods as well as a comparison process for categorization methods.

We would like to conclude this chapter by emphasizing again that categorization is not fundamentally different from feature transformation. Both steps aim at data reduction, the latter generally, the first for some context. The categorization process can be divided into a micro process of pair-wise comparison of descriptions (some object against the context) and a macro process for the management of the entire media database. In the Chapters 11 and 21 we will provide an in-depth analysis of the components and steps of the various categorization approaches. We will endeavor to link the steps to those taken in feature transformation.[1] The result will be that, actually, the methods are related in several respects.

The methods discussed in this chapter have in common that all of them employ rather simple models. Most of them are heavily dependent on the dimensionality of the employed data. That is, the performance of these methods

[1] That is, we apply a media understanding process on the methodology of media understanding.

tends to fall over-linearly with increasing dimensionality. In the next section, we encounter methods with different characteristics: complex models, costly training, excellent performance: the probabilistic categorization methods.

Chapter 9

Probabilistic Categorization

Introduces fundamental concepts of probabilistic inference, discusses independence-based models such as the Bayesian classifier, explains the general concepts of Bayesian networks and their application in Markov processes.

9.1 Foundations of Probability Theory

The categorization approaches discussed so far relied on given information: the model, references or ground truth. If the given data represents the media understanding problem well, the performance will be good. If not, then not. In this chapter, we introduce a new concept. We employ statistical methods, namely probability theory, on the given data in order to come to a better understanding of the input data and, eventually, of the media understanding process.

This chapter summarizes *probabilistic* approaches to categorization. In the first section, we introduce all required concepts (e.g. Bayesian inference) and discuss relevant practical issues such as sampling of probability densities. The second section deals with probabilistic categorization methods that rely on the assumption that description elements are independent of each other. The third section goes one step further by allowing *conditional dependencies* between description elements, which results in the general theory of *Bayesian networks*. In the last section, we introduce several Markov processes as examples for particular types of Bayesian networks that have proven successful in media categorization.

Figure 9.1 sketches the general probabilistic categorization problem. It is not fundamentally different from the categorization model introduced in the last chapter. Again, we have a training step in which the probabilistic model, essen-

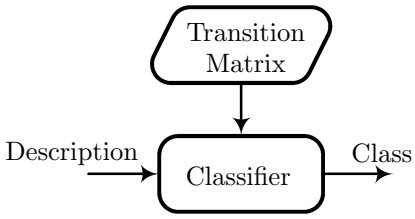


Figure 9.1: Probabilistic Categorization.

tially a transition matrix for all possible events, is computed from training data. The model is applied on descriptions in the application step. The result is non-surprisingly some class labeling. As we will see below, training of probabilistic models can be a tedious task. The application, however, is usually very efficient.

Probabilistic models are often visualized as graphs where the nodes represent states and the edges transitions between states. Figures 9.2 and 9.4 are examples of graph representations. The outer form is the same as in state transition graphs. The inner meaning, however, is slightly different. While in the case of state transition graphs, the transitions are deterministic and triggered by certain events, in probabilistic graphs the transitions are only described statistically. That is, in m of n cases the transition happens, but it cannot be predicted if the transition happens in one particular situation.

For calculations such as those in a categorization process, the graph view is not efficient. Then, the knowledge of a probabilistic classifier can be expressed in superior form as a series of matrices of all events that may occur jointly. For example, in a situation where we have two binary events x and y with the probabilities $P(x) = 0.3, P(y) = 0.8, P(x|y) = 0.2$ the table of *joint probabilities* looks as follows:

	x	$\neg x$	*Sum*
y	0.16	0.64	0.8
$\neg y$	0.14	0.06	0.2
Sum	0.3	0.7	1

Table 9.1: Matrix Representation of a Probabilistic Network.

The joint probability of events x, y occurring together is computed by the chain rule: $P(x, y) = P(x|y).P(y) = 0.16$. In the context of media understanding, x may stand for a condition of the form $f_1 < \epsilon_1$, i.e. some feature f_1 is smaller than a threshold. Event y may be defined similarly. The values of the table can then be interpreted as the likelihoods of particular classes associated

with the cells of the table. In the section on Bayesian networks, we will discuss the usage of such matrices for categorization. Furthermore, we will generalize the idea of deriving *probabilistic events* from description elements x, y by conditions such as $m(x, y) < \epsilon_i$.

Probabilistic categorization models are *inference models*. That is, general knowledge about the world (the *a priori*, expressed, for example, as ground truth) is combined with some new facts (e.g. the description of a query) in order to arrive at a refined view of the world (the so-called *a posteriori*). Probabilistic inference transforms a priori into a posteriori, practically, given descriptions into weighted descriptions. In the context of media understanding, *refined* may mean adapted to a particular class of problems, a particular query, etc. Inference from a priori to a posteriori is also called *forward reasoning*. Despite the temporal connotation, often, *backward reasoning* is possible – typically employed for improving the given references or ground truth.

The introductory section must also repeat the fundamental rules of probabilistic calculation. The following math block summarizes all rules:

$$P(x = n) = \frac{\text{number of times, event x=n happens}}{\text{number of times, event x=anything happens}} \tag{9.1}$$

$$P(x, y) = P(x|y).P(y) = P(y|x).P(x) \tag{9.2}$$

$$\Rightarrow P(x|y) = \frac{\text{number of times, events x and y happen together}}{\text{number of times, event y happens}} \tag{9.3}$$

The first line just defines simple probabilities. Remark on notation: For the sake of simplicity, below we write $P(x)$ where we mean $P(x = n)$ if n can be considered apparent. The second line expresses actually two rules. The first equation is the *chain rule* which expresses that joint probabilities such as $P(x, y)$ can be expressed by *conditional probabilities* such as $P(x|y)$ and a priories $P(y)$. The second equation is the famous Bayes theorem:

$$P(y|x) = P(x|y).\frac{P(y)}{P(x)} \tag{9.4}$$

Bayes theorem is the foundation of Bayesian inference, where $P(x|y)$ is the world knowledge which is weighted by the a priories in order to arrive at the a posteriori $P(y|x)$. Very simple, but very effective as we will see in the next two sections. Eventually, the third line expresses the meaning of conditional probabilities textually. It is normalizing a joint probability by its a priori part.

Bayes theorem is the major asset of all probabilistic inference methods. Its particular strength lies in the fact that it is partially against human intuition. As the authors of [182] could show most humans do not take a priories into

account in their reasoning. Two famous examples should illustrate this point well.

First example: Game show. The scenario is a game show where the candidate must choose one of three doors and wins what is behind this door. There is a car behind one door and goats behind the two others. The proceeding is always that the candidate selects one door, the showmaster opens a different door behind which a goat becomes visible and asks the candidate to reconsider. The question is: Should the candidate change her mind?

The answer given by most people confronted with this problem is no – there are no new facts. However, Bayes theorem says something else. If $P(x)$ is the a priori probability that the car is behind door x, i.e. $P(x) = \frac{1}{3}$, and $P(y)$ is the probability that one of the two doors not chosen by the candidate remains closed, i.e. $P(y) = \frac{1}{2}$ then $P(x|y) = P(y|x).\frac{P(x)}{P(y)} = \frac{2}{3}$. That is, the probability that a car is behind the door which remains closed is two thirds, because $P(y|x) = 1$, i.e. the showmaster would never open the door with the car behind it.

Second example: HIV test. Here, the question is: How likely is it that a US citizen is HIV positive if his HIV test is positive? Let $P(x)$ be the probability that the test is positive and $P(y)$ be the a priori probability that the candidate is HIV positive. In the US $P(y) = 0.01$, i.e. only about one percent of the population is HIV positive. Let the first degree of error (test is negative but the person is HIV positive) be $\alpha = 0.05$. Then, $P(x|y) = 0.95$. Furthermore, let the second degree of error (test is positive if the person is not ill) be $\beta = 0.01 = P(x|\neg y)$. Then, $P(x) = 0.0194$ and $P(y|x) = 0.49$ according to Bayes rule. That is, the probability is less than fifty per cent!

In conclusion, Bayes weighting is a very strong mechanism of transforming a priori probabilities into context-specific a posteriori probabilities that should be employed for the benefit of better categorization.

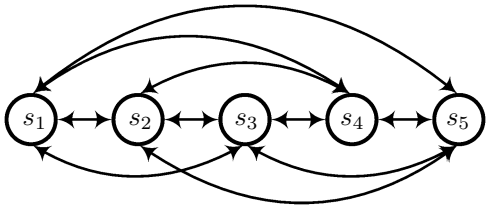


Figure 9.2: Gibbs Sampling.

In order to do that we have to solve the – central – *sampling* problem: the provision of the density functions $P(x)$ and the conditional probabilities $P(x|y)$. This can only be done if a fraction of reality is observable. Then, two major methods are available.

- Gibbs sampling
- Expectation maximization

Gibbs sampling *simulates* the probabilistic process that should be described and aggregates the densities from the simulation output. Figure 9.2 shows an example. For the given training data, the five-step process is iterated and by doing that values for $P(s_i)$ and $P(s_i|s_j)$ are aggregated. That is, the sequence of occurrence of the events matters. The quality of the result depends on the quality of the training data. Gibbs sampling is an example for a Markov process. The construction of densities can be characterized as probabilistic.

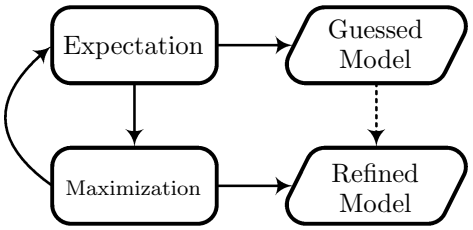


Figure 9.3: Expectation Maximization Algorithm.

Expectation maximization, on the other hand, is a deterministic process. Figure 9.3 shows the two steps and their respective results. The density functions are modeled as functions $f(x, a)$ where a is a parameter vector. For example, f may be a normal distribution and $a = (\mu, \sigma)$. The algorithm employs the following steps:

1. Estimate the parameter set a. The first guess may be random.
2. Compute the gap between the resulting f and the training data. Adjust the parameters a accordingly.
3. Return to the first step until the gap is smaller than a pre-defined threshold.

Hence, expectation maximization employs a rigid model that is adapted to the given evidence (training set). This method is less flexible than Gibbs sampling but less prone to overfitting to the training set.

Overfitting is a general problem of density estimation. One popular method to limit the effect of the training data on the estimation process is *Monte Carlo sampling*. The idea is very simple. Instead of the actual training data x, we employ randomized data $r(x)$ – r being the randomization function. If r is chosen appropriately the iterated process may eliminate an undesired bias in the training data. Typically, r may be Gaussian-like function that distributes

the measurements in the form of the normal distribution. This randomization causes movement towards some pre-defined mean. Monte Carlo sampling can be combined with both Gibbs sampling and expectation maximization. Since both methods aggregate the input of the entire training set it is guaranteed that the randomization function does not introduce white noise. The practical side of sampling will be touched in the next chapter.

Before we dive into the pool of probabilistic methods, we would like to emphasize some major differences between deterministic methods (like those discussed in the last chapter) and probabilistic methods.

1. The type of model is fundamentally different. Deterministic methods work directly on the data and are able to compute quantities as well as predicates. Probabilistic methods require some pre-processing step that transforms quantities into *proto-predicates*. Since this transformation is already a categorization process, probabilistic methods can be viewed as operating on a meta level.

2. Deterministic classifiers base their decisions on similarity to references (e.g. k-NN, k-means) or the maximum distance of references (such methods will be discussed in the second part). They hardly take a priories into account. Probabilistic methods base their decisions on weighting of given facts by a priories which produces a statistical truth that may be perceived as counterfactual by humans.

3. Deterministic methods employ training data mostly in a separative way. That is, description elements are compared to corresponding ones but no co-occurrences are considered. This is only partially true for dynamic association models which, in return, require optimization algorithms that make them less deterministic (danger of local optima). Probabilistic methods per se emphasize joint events without a need for optimization.

In conclusion, probability theory is a promising starting point for the development of categorization methods that take correlations of events (descriptions) into account. The major practical limitation is the provision of expressive probability densities. If this problem can be overcome, Bayesian inference models such as those discussed in the next two sections can be employed for successful media understanding.

9.2 Independence-based Categorization

The classifiers discussed in this section have in common that they assume joint probabilities $P(x, y)$ to be the product of components $P(x, y) = P(x).P(y)$.

That is, the components x, y are independent of each other. Since this assumption is quite unrealistic these methods are called *naïve*.

In media understanding terms, this means that we assume the description elements to be independent of each other. Since this is hardly ever the case in one of the methods discussed in the first part of this textbook, it is always recommendable to run a factor analysis before independence-based categorization. Factors are linearly independent of each other which – at least in the mathematical sense – justifies considering them as semantically unrelated.

Below, we focus on two approaches: the *Bayesian classifier* (BC) and the *binary independence model* (BIM). The BC is a simple form of a Bayesian network (see next section) while the BIM is a specialized form of a probabilistic model for information retrieval. Both methods employ Bayes theorem and are very efficient.

The Bayesian classifier is not necessarily a naïve method. It is just the frequent practical application that associates BC with the independence assumption. The BC categorizes the input data $x \in F$ in $c_i \in C$ classes by the following rule:

$$BC(x) = c_i \text{ with } i = \arg\max_c P(c_i|x_1 \wedge x_2 \wedge .. \wedge x_n) \tag{9.5}$$

Applying Bayes theorem the probability term of the criterion can be transformed to the following a posteriori:

$$P(c_i|x_1 \wedge x_2 \wedge .. \wedge x_n) = P(x|c_i).\frac{P(c_i)}{P(x)} = \frac{P(x|c_i).P(c_i)}{\sum_j P(x|c_j).P(c_j)} \tag{9.6}$$

That is, the maximal conditional probability of descriptions and classes is the winner (*maximum likelihood principle*). Building the conditional probabilities $P(x|c)$ requires sampling from a given ground truth (samples + labels). In particular, for every class samples of all possible combinations of description elements should be given. Such an estimation process would be called a *joint density estimator*. Since a sufficiently large ground truth is hardly ever available, the naïve BC assumes the elements of x independent: $P(x|c) = \prod_j P(x_j|c)$. Then, only pairs of description elements and classes $P(x_j|c)$ have to be sampled, which can be done by simple Gibbs sampling.

Alternatively, the conditional probabilities $P(x_j|c)$ can be assumed to be Gaussian and expectation maximization can be applied. In this case, the sampling problem is shifted to identifying the best parameters of the normal distribution for the given ground truth data. Such a Bayesian classifier is called a *Gaussian Bayesian classifier*. It is still naïve but less close to the ground truth than the Gibbs sampled classifier.

Recently, efforts have been undertaken to limit the negative effect of the independence assumption in categorization by the BC. The *random fern* approach suggests not to assume all elements of the input data as independent but to compute joint densities for the most important elements. Here, *most important* can be operationalized as semantically related, for example. The introduction of this idea opens a continuum between the poles of independence assumption and joint density estimation. Hence, depending on the ground truth the quality of the results should also be somewhere between the given extremes.

Practically, the BC is in all its forms a very popular classifier. The usage of Bayesian inference in a simple model provides an effective tool for quick categorization. The major limitation is the independence assumption which is, in particular, in the field of media understanding, simply unrealistic. We recommend using the BC for the first orientation in the media understanding process. Feeding a BC with a small ground truth shows the potential of a particular set of feature transformations. Even for the best categorization methods it should usually not be possible to outperform the Bayesian classifier by too far.

The binary independence model was originally developed for text information retrieval [109]. It assumes input descriptions with binary predicates as elements. These elements are usually interpreted as presence/absence values of terms in a document (*predicates*). However, BIM could in the same manner be used with any other interpretation of the descriptions.

The goal of BIM is to rank the members of a media database by relevance to a given reference (in retrieval, a query object). In the following, we denote the description of the reference as f and the description of one object as x. BIM computes a *retrieval status value* (RSV) for any x, ranks the media objects by the RSV and assumes the first n objects as relevant and the rest as not relevant. That is, BIM performs a binary categorization.

The RSV is computed from the odds that a particular x is relevant to r against x being not relevant, i.e. relevant to $\neg r$. The derivation goes as described in Equation 9.7.

The first line defines the odds of relevance against irrelevance. Then, we apply Bayes rule and remove weights that appear twice. From second to third line we get by assuming independence of the description elements x_i. Then, we split the product in two by distinguishing between present predicates ($x_i = 1$) and others. The first term remains unaltered in the final formula. The second, however, is transformed using two assumptions. Since RSV is only used to order media objects, all terms that do not exist in the reference ($r_i = 0$) are ignored. Secondly, for the rest, the definition $P(x = 0|r) = 1 - P(x = 0|\neg r)$ is stated. That is, if a term does not exist, we assume that it may be as likely relevant as irrelevant. These assumptions allow for the merging of the two products.

$$
\begin{aligned}
RSV(x) &= \frac{P(r,x)}{P(\neg r,x)} = \\
&= \frac{P(x|r).P(r)}{P(x|\neg r).P(\neg r)} = \\
&= \prod \frac{P(x_i|r)}{P(x_i|\neg r)} = \\
&= \prod_{x_i=1} \frac{P(x_i|r)}{P(x_i|\neg r)} . \prod_{x_i=0} \frac{P(x_i|r)}{P(x_i|\neg r)} = \\
&= \prod_{x_i=r_i=1} \frac{P(x_i|r_i)}{P(x_i|\neg r_i)} . \frac{1-P(x_i|\neg r_i)}{1-P(x_i|r_i)}
\end{aligned} \tag{9.7}
$$

Practically, $\log RSV(x)$ is used for ranking, because then the product is replaced by a sum which can be computed faster. The BIM was developed for ranking of text documents [109]. Still, the mechanism is applicable to all media descriptions that employ predicates (for example, in media understanding of media understanding). Then the usage is limited to situations where the relevance/irrelevance of description elements has a semantic meaning (as in text understanding). So far, the BIM has been widely neglected outside the information retrieval domain. We believe that this method may be an interesting alternative to distance-based methods employed today in content-based audiovisual retrieval.

The major difficulty of using the BIM is – as always in probabilistic categorization – the sampling of the $P(x|r)$ values. The last step of the derivation is primarily performed for simplifying this problem, since it reduces the number of required density values. Thanks to the independence assumption, training a binary independence model requires only a small ground truth – with the downside that the independence assumption is not realistic in media understanding.

In conclusion of this section, we have introduced two frequently used probabilistic models that employ Bayes theorem in order to get from an a priori situation to a contextually relevant a posteriori. Both methods assume description elements to be independent – which is mostly not the case in reality, but helps overcoming the problem of density estimation. The effect of BC and BIM on dimensionality and, hence, performance of the media understanding process is very positive. Very high-dimensional feature spaces can be processed efficiently. On the other hand, the training requires a well-balanced ground truth which is hard to define. In the next section, we will introduce a generalized model of Bayesian inference that can be employed on arbitrary data.

9.3 Bayesian Networks

Bayesian networks (BN, also known as *probabilistic networks*, *belief networks*) are graphical models with a directed flow of information.[1] BN is a *modeling technique* for complex probabilistic systems. The big advantage of BN is that the complexity of categorization is transferred from the actual reasoning to the modeling. After the model has been established the calculations are always the same.

Below, we discuss the structure of Bayesian networks and their application for media understanding. BN is a general approach, categorization in media understanding just one type of BN application. In the course of discussion, we introduce the graph model, the matrix representation of BN as well as the rules of calculation and the conditional dependence problem which is the central element of complexity in BN theory.

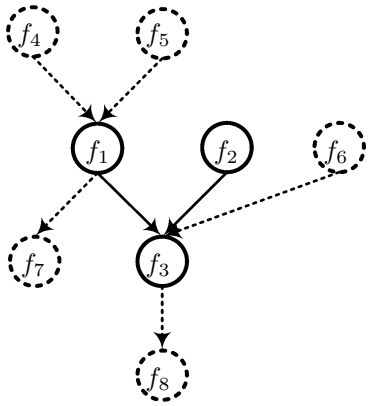


Figure 9.4: Bayesian Network Example.

Figure 9.4 illustrates a typical Bayesian network. Nodes stand for events (in our terminology, description elements in the form of predicates or decision rules). Arrows stand for transitions. The flow of information (inference) goes from top to bottom. Central to BN modeling are dependent nodes. The two nodes f_1, f_2 influence the state of f_3, i.e. $P(f_3|f_1, f_2) \neq 0$. Such probabilities are called *conditional probability distributions* (CPD) and usually expressed in the form of Table 9.1. Depending on the context of usage, such matrices are also called *confusion matrices*, because they express the probability that two dependent events are confused with each other (similar to the odds in the BIM).

[1] An example for a probabilistic model with an undirected flow of information would be the Markov random field model (see next section).

It is worth noting that the size of the CPD matrices grows exponentially with the number of dependencies, i.e. the complexity of the problem. Therefore, the most important criterion of complexity in BN modeling is *conditional independence* of nodes, and wise BN modeling will aim at avoiding dependencies wherever possible – of course without oversimplifying the problem.

Bayesian networks theory knows two types of reasoning:

- Forward reasoning (predictive reasoning)
- Backward reasoning (diagnostic reasoning)

In the first case, the graph is traversed from top to bottom and the likelihood of dependent nodes is computed. Typical applications are the computation of the *overall likelihood* of a model (net) or the likelihood of a result given an assembly of events. Backward reasoning tries to detect the cause for a (certain) effect. It is typically used for reconstruction of the assembly of events that caused a particular result. In media understanding, forward reasoning is the more relevant application, as we will see below.

Now, we use the graph in Figure 9.4 to explain the rules of calculation used in BN. All calculations are based on Bayes theorem, which reads for three connected events f_1, f_2, f_3 as follows:

$$P(f_3|f_1, f_2) = \frac{P(f_1, f_2, f_3)}{P(f_1, f_2)} = \frac{P(f_1, f_2|f_3).P(f_3)}{P(f_1, f_2)} \tag{9.8}$$

The joint probability of the subgraph f_1, f_2, f_3 can be computed by simply applying the chain rule.

$$P(f_1, f_2, f_3) = P(f_3|f_1, f_2).P(f_1).P(f_2) \tag{9.9}$$

Computing the joint probability is a typical case of forward reasoning. If only partial data is available for reasoning, variables have to be introduced for the missing events. For example, if we want to know whether f_3 was caused by f_1 but we have no information about f_2 we have to iterate over all possible values of this event:

$$P(f_3|f_1) = \sum_y P(f_3|f_1, y).P(f_1).P(y) \tag{9.10}$$

If no input is given, the probability of f_3 can be computed by adding another variable:

$$P(f_3) = \sum_x \sum_y P(f_3|x, y).P(x).P(y) \tag{9.11}$$

Backward reasoning can be performed in the same manner by applying Bayes rule. Nodes like f_6 can be treated like f_1. Nodes like f_4, f_5 require the application of the chain rule, i.e. their integration in the CPD of f_1. Eventually, nodes like f_7, f_8 are only dependent of their predecessors and have to be modeled by the chain rule.

In conclusion, reasoning in Bayesian networks requires only two tools:

1. Application of the chain rule for dependent nodes. Implicitly, this includes the application of Bayes theorem.

2. Representation of missing data by variables and summarization over all possible manifestations of missing events.

As we stated above, the rules of computation in BN are always the same. The problem is modeling an appropriate network. That is, the definition of events and dependencies and the sampling of the CPD. For the latter task, the density estimators discussed above can be used. Since the top layers (so-called *hierarchical priors*) have a high influence on the BN reasoning, they have to be sampled with great care. The high dependency on the hierarchical priors is one of the largest weaknesses of BN. Eventually, the modeling has to be done as required by the problem domain. Below, we will discuss the typical requirements of media understanding.

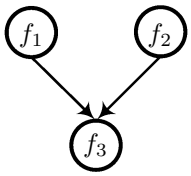


Figure 9.5: Conditional Dependence.

Generally, the central elements of BN are *conditional dependent* nodes. Figure 9.5 shows two nodes f_1, f_2 that are conditional dependent. Conditional dependence means that both events f_1, f_2 can be used to explain the result f_3. However, as soon as we know that one of them is the actual cause of f_3 the other one becomes less likely. This phenomenon is called the *explaining away paradoxon*. The explanation is actually very simple. The chain rule sets the following equivalences between conditional probabilities:

$$P(f_3) = P(f_3|f_1).P(f_1) + P(f_3|f_2).P(f_2)$$

That is, event f_3 can be explained by f_1 or by f_2. As long as we do not know the actual cause of the outcome, the conditional probabilities will be smaller

than one. As soon as the cause is known, one becomes maximal and, therefore, reduces the other conditional probability. For example, if we come to know that f_2 caused f_3 ($P(f_3|f_2) = 1$) the above formula becomes:

$$P(f_3|f_1).P(f_1) + 1.P(f_2) = \text{const.}$$

Since the a priories remain constant, $P(f_3|f_1)$ must necessarily be explained away a bit.

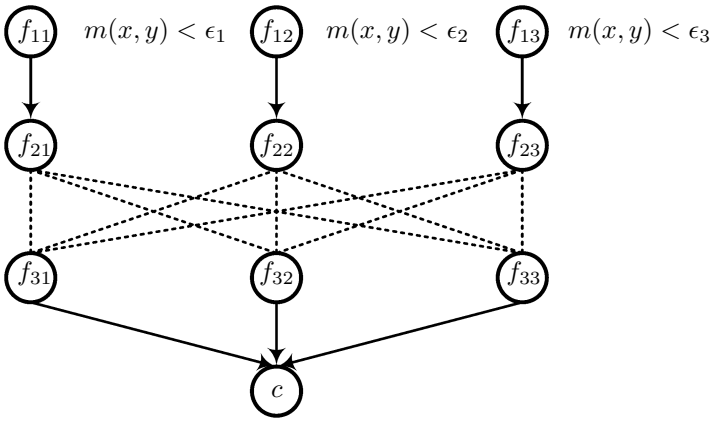


Figure 9.6: Bayesian Networks for Media Understanding.

The Bayesian network model is nice, but how can it be employed for categorization in media understanding? Figure 9.6 illustrates the concept. First of all, we regard description elements and class labels as events, the first located on the top of the network, the latter on the bottom. Predicates fit naturally with the event concept of BN. However, if quantities are given, we require a pre-processing step in which the quantities are transformed to *proto-predicates*. In the figure, we suggest a distance-based approach, i.e. measurement from description element x to a reference y. In practice, often rule-based approaches are employed. Of course, coarse representation would be another option, etc. As soon as the proto-predicates have been computed the second round of categorization can be performed. In the simplest case, the class label can be computed directly from all inputs. Since this model would create many dependencies, it may make more sense to add hidden layers that combine related description elements before reasoning on the class. For example, if audio and color features are given, they may be processed separately first before the categorization in videos of, say, sports events or newscasts is performed.

This approach may give the impression that the design of an appropriate BN for media understanding depends heavily on the experimenter. However, in

practice a different approach is usually taken. The experimenter only provides a ground truth and in the training step some meta-algorithm computes the best BN for its representation by trial and error.

In conclusion, the Bayesian networks model is a formalism that fits all probabilistic categorization problems. After the modeling of the problem, the categorization process (forward reasoning by the chain rule using variables) is always the same. Against its elegance and simplicity stands the low efficiency of BN. The preparation of the model and the inference process require more resources and time than most other categorization approaches. The effect of BN is positive in terms of reducing the semantic gap (application of Bayes rule) and dealing with noisy data (conditional dependence improves the chance for correct reasoning) but bad in terms of dependency on a large well-balanced ground truth.

In the last section of this chapter, we introduce Markov processes which are today successfully employed for categorization in media understanding and other domains.

9.4 Markov Processes

Markov processes are particular types of Bayesian networks. They are also modeled as graphs and the elements are the same: Nodes stand for events and arrows for probabilistic transitions between events, i.e. conditional dependence. The usage, again, is the same: Estimation of the probability of occurrence of an observed sequence of events. The big difference between Bayesian networks and Markov processes, however, is that the latter have a fixed, pre-defined structure. Therefore, the problem of *designing the network* falls away in categorization by Markov processes. Markov processes with distinct structures are known by different names: hidden Markov models are one prominent example. By stating that one Markov process has a particular structure, we do not mean the number of nodes would be fixed (it is not) but that the interpretation of particular nodes is always the same and the relationship of differently interpreted nodes is always the same. The examples given below will illuminate this point.

In the remainder of this section, we discuss three types of Markov processes: Markov chains, hidden Markov models and Markov random fields. The latter type is not a 'typical' Bayesian network and only of limited interest in probabilistic categorization, but they are sometimes employed in image processing and then, very effectively. Therefore, we consider it beneficial to explain this approach briefly here.

Figure 9.7 gives an example of a Markov process of first order. Such a Markov process is called a *Markov chain*. The nodes represent the events. Every event depends only on the first predecessor (not on pairs, etc. of predecessors), i.e. order $n = 1$. Please note that transitions from one node to itself (that is, one

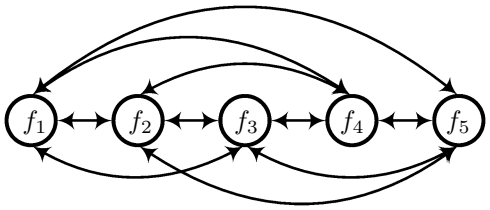


Figure 9.7: Markov Process Example.

event causes the same event) are omitted in the figure. This was done for the sake of simplicity. A full Markov chain will also allow a flow from one node to itself.

Due to the chain-like structure of Markov processes, they are often employed in a temporal context. Then, the transitions between events imply progression in time. Therefore, the picture given in Figure 9.7 provides only a static view of the problem but does not tell anything about the temporal structure of the problem. It is important to keep this distinction in mind in order not to get confused by the different ways of illustrating/modeling Markov processes.

Simple Markov processes such as Markov chains are only of little significance in media understanding. They are usually not employed for categorization because the one-layer structure does not offer a sufficient number of degrees of freedom for modeling all aspects of the media understanding categorization problem.

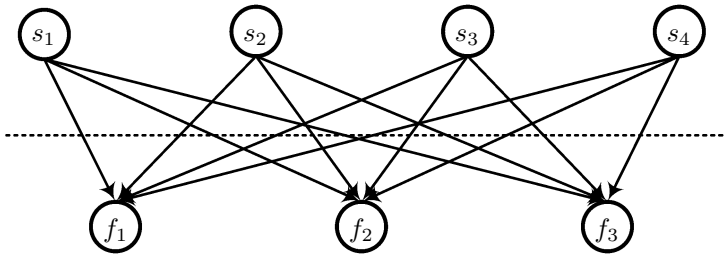


Figure 9.8: Hidden Markov Model Structure.

Instead, an extended form of Markov process known as *hidden Markov model* (HMM) is employed. The HMM is based on the assumption that the actual Markov process is not directly observable. That is, we cannot say – in the temporal context – which is the current state and which will be the next. Rather, the HMM introduces additional *observations*: states, that are probabilistically linked to the states of the process and observable.

Figure 9.8 shows the static structure of a HMM with three *hidden states* f_i and four observations s_i. A typical toy example used in lectures to explain HMM is estimating the weather behind a mountain (hidden states) from the weather at the observer's location (observations). Why are HMM are so powerful in media understanding? The answer is straightforward. Media understanding aims at understanding content from descriptions. Between the two phenomena lies the semantic gap. If we consider the media content *hidden* but the descriptions *observable* we arrive at the situation for which HMM were designed. It is, therefore, only natural to employ them for media understanding categorization.

In order to build a HMM we need the following information:

- Hidden states and observations. Below, we use F for the set of hidden states f_i and S for the set of observations s_i.
- Inference matrices between observations and hidden states *and* between hidden states. The first is clear. The second type of inference should not be forgotten, though. HMM model a hidden Markov process of first order. Therefore, we have probabilistic transitions between hidden states.
- A vector Π of probabilities π_i for the likelihood that the initial state is f_i. Since we cannot observe the Markov process, we need at least a guess for its a priori state.

One actual challenge of using HMM is providing the confusion matrices of observations and hidden states. The number of observation states may be smaller, equal or larger than the number of hidden states. Of course, the quality of inference depends on the number of observations (generally, the more the better).

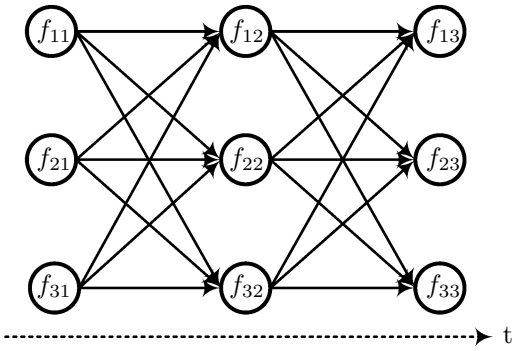


Figure 9.9: Reasoning in Hidden Markov Models.

When the required data has been sampled or otherwise provided the actual inference follows the process illustrated in Figure 9.9. It is common to

assume a temporal context in the reasoning process. That is, the probability of a state f_{it+1} depends on all its predecessors f_{it} and their transition probabilities. Furthermore, the observations – not illustrated in the figure – are taken into account. For the media understanding application, the temporal context can be understood in the following way. The HMM starts with an initial guess of the categorization given by Π. This guess is refined over time by transforming a priories into a posteriories by adding one description element (or a group of elements) per time step. Eventually, we can derive the best possible categorization from the probabilities computed in the HMM.

Taxonomically viewed, hidden Markov models can be used to solve three problems:

1. *Evaluation*: Identify the best HMM for given observations. In evaluation, one HMM stands for one class. The entire categorization problem is modeled by a set of HMM. A typical application is speech recognition where one HMM stands for one word, and the observations are, for example, given as temporal audio descriptions (zero crossings, etc.). The evaluation problem is typically solved by the *forward algorithm*.

2. *Decoding*: Identify the most likely sequence of hidden states for a given sequence of observations. Here, one HMM covers the entire categorization problem, and the selection is performed by the maximum likelihood principle. A typical example is computational linguistics where the words of a phrase (observations) are labeled as nouns, verbs, etc. (hidden states). The decoding problem is typically solved by the *Viterbi algorithm*.

3. *Learning*: Compute the HMM contents (confusion matrices, initial states) from given observations and hidden states. This problem is typically solved by applying evaluation and decoding alternatingly in an expectation maximization process until the match between observations and predicates is sufficiently good.

For many media understanding applications, the forward algorithm is of greatest importance. It computes the overall probability of a HMM by the following two equations:

$$\alpha_{j1} = \pi_j . P(f_j|s_1) \tag{9.12}$$

$$\alpha_{jt+1} = \sum_{i=1}^{n} \alpha_{it} P(f_j|f_i) P(f_j|s_{t+1}) \tag{9.13}$$

The α_{it} are bound variable that accumulate the a posteriori probabilities, n is the number of hidden states. The two types of confusion matrices are expressed

in the conditional probabilities. The start probabilities α_{i1} depend only on the initial guesses and the inference from observations to hidden states. From the second iteration on $(t+1)$ the prior probabilities are summed up and weighted by the transition probabilities of the hidden states and, again, by the inference from observations to hidden states. Eventually, the overall probability of the HMM is $P(F,S) = \sum_{i=1}^{n} \alpha_{iT}$, where $T = size(S)$.

The forward algorithm is a straightforward optimization of this process. Instead of implementing two loops it accumulates the a posteriori probabilities in a recursive procedure which transfers parts of the complexity from the algorithm to the data (on the stack) and reduces the order from $O(n) = n^2$ to $O(n) = n.log(n)$ – a strategy well known to every graduate computer engineering student.

The Viterbi algorithm implements the same idea and uses equations that are very similar to the forward algorithm. The only difference is that in decoding we are interested in the maximum likelihood instead of the overall likelihood. Therefore, the Viterbi algorithm replaces the sum by the maximum operator:

$$\beta_{j1} = \pi_j.P(f_j|s_1) \tag{9.14}$$

$$\beta_{jt+1} = \max_i\big(\beta_{it}P(f_j|f_i)P(f_j|s_{t+1})\big) \tag{9.15}$$

Eventually, the class label of the hidden state with the maximum likelihood is attached to the media object under investigation.

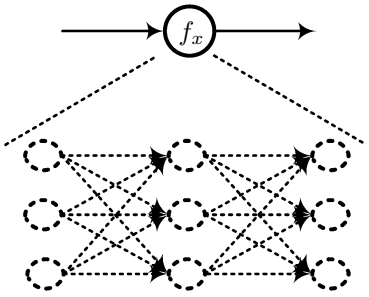


Figure 9.10: Hierarchical Hidden Markov Models.

Hidden Markov models are very effective probabilistic classifiers. However, if their structure is insufficient to cover complex categorization problems, they are sometimes replaced by *hierarchical HMM*. Figure 9.10 illustrates the structure. Simply, every hidden state may be replaced by a HMM, and so on. Since this recursion may go to an arbitrary depth hierarchical HMM are theoretically

very powerful. In practice however, hierarchical HMM are very hard to train. One rule of the thumb is that *for training of an expressive HMM at least ten ground truth samples per hidden state are required.* With hierarchical HMM, an exponentially bigger ground truth would be required. Since furthermore, every hierarchical HMM can be reduced to a HMM, hierarchical HMM are not to widely used in media understanding.

Markov random fields (MRF) are related to Bayesian networks. The only difference is that in an MRF, the transitions are not directional, i.e. information may flow from any node to any other connected node and back. Since the edges of the MRF graph have no direction, Bayesian inference is impossible. Therefore, MRF are unsuitable for categorization. Instead, MRF are employed in media understanding, for example, for image segmentation where each node represents a (group of) samples and the probabilities model the likelihood of edges between samples. The essential problem of MRF usage is computing the transition probabilities. Solving it requires an iterative process of annealing. The Hopfield network (Chapter 26) may be seen as one example of such a process.

Where are Markov processes employed in media understanding? As already mentioned, hidden Markov models are of highest significance for categorization in media understanding. HMM are the state-of-the-art solution for speech recognition and, for example, employed in bioinformation processing for motif identification. In speech recognition, the forward algorithm is used to solve the evaluation problem. That is, the problem of categorization is shifted from probabilistic inference to *quickly identifying those HMM that are most likely to represent a certain speech description.* This problem can be solved by wrapping another media understanding process around it, applying a feature transformation on the HMM data and categorizing the resulting descriptions. Practically, the HMM data are often abstracted to a simple data vector, and categorization is performed using the vector space model. This approach is straightforward and something more sophisticated would be imaginable. Why not applying a decoding HMM?

In bioinformation, motifs are short sequences of base pairs (5-20) that appear several times in transcription factors bounding sites (TFBS) which are responsible for the transformation of DNA into proteins. One of the most important problems of bioinformation understanding is the identification of motifs. One typical solution is training hidden Markov models by Gibbs sampling or expectation maximization for the identification of such sequences.

In conclusion, Markov processes are specialized Bayesian networks with predefined structure that can be applied very effectively. Their application is state-of-the-art in several areas of media understanding because they reduce the semantic gap problem by Bayesian inference, have a very good performance and are less prone to noise than many deterministic categorization methods. On the other hand, the training process requires a well-balanced ground truth of

considerable size. Furthermore, the application of an iterative sampling model is very time-consuming.

The combination of sophisticated training and fast application makes Markov processes – as well as the other probabilistic categorization methods – tailor-made for mobile application where the training is performed on the desktop while the usage is performed mobile. In the next chapter, we will go deeper into the practical side of media understanding and discuss topics such as application design, implementation and evaluation.

Chapter 10

Application Building

Introduces the media understanding application design process, requirements of matching, retrieval and browsing, aspects of different programming environments, methods and measures for evaluation as well as query acceleration approaches.

10.1 Application Design

This chapter illuminates the practical side of media understanding. It is of paramount importance to know the algorithms of feature transformation, information filtering and categorization that are used in media understanding, but it is also relevant to know a handful of best practices for the implementation of media understanding applications. Below, we introduce major concepts of application design and implementation as well as for testing and refinement. The four sections follow the classic waterfall model of software engineering: design, implementation, evaluation and optimization – though the media understanding software engineering process is not that old-fashioned. The practical implementation will follow a test-driven approach where task sand evaluation criteria are defined first, and the proper algorithms are chosen based on the given requirements. Media understanding software design is an iterative process. The first prototype is refined based on evaluation results until the initial requirements are met as good as possible.

We firmly believe that media understanding application design cannot be left to practitioners while the researchers focus on algorithms for description and categorization. It is, in the contrary, an integral part of the research process. Media understanding is a practical discipline where the development of

theoretical models, implementation and parameter tuning go hand in hand in solving a problem. The media understanding researcher is therefore encouraged to approach this research area from the practical side. Only the persevering endeavor to solve a particular media understanding problem can trigger the learning process that makes one understand what the real difficulties of media understanding lie.

The remainder of this section is organized as follows. First, we introduce the major types of applications of media understanding. Then, we sketch the typical requirements of such applications, outline the design process for desktop applications and discuss a few design patterns – including the media understanding of media understanding scheme. Eventually, we emphasize the particulars of mobile media understanding since the relevance of this area of engineering is increasing today.

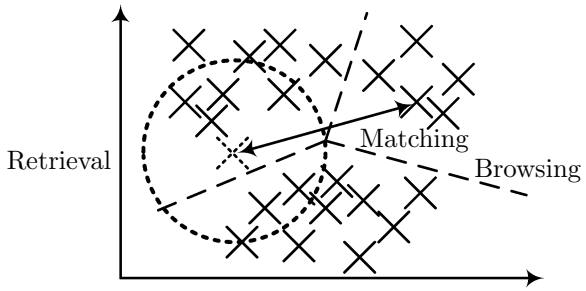


Figure 10.1: Matching, Retrieval and Browsing.

Figure 10.1 uses a feature space to illustrate the three major types of applications of media understanding. *Matching* as the simplest form of media understanding, tries to identify the *same content* in a database as given in a reference. In the figure, the dotted x is the reference and the double-arrowed line indicates the match. *Retrieval* aims at identifying the n media objects that are most similar to the given reference, in the figure illustrated by the dotted circle. Please note that the reference may be given as a media object, a group of media objects or a description. The diameter of the circle depends on the location of the reference as well as on the density of feature space. The third type of application, *browsing* aims at separating clusters (classes) of semantically related media objects and, optionally, at quantizing the level of similarity between clusters. The dashed lines indicate the clusters in the figure.

Each practical application can be categorized as either matching, retrieval or browsing. Face recognition is, for example, a typical matching application. Content-based image search is typically implemented as a retrieval application (see Figure 10.5 for an example). Music genre classification is an example for

browsing. Here, semantically similar content is clustered in genres such as classic music, pop, jazz, etc. If necessary, related clusters such as rock music and heavy metal can be organized hierarchically by similarity.

The type of application determines the *design requirements*. Matching applications have to provide one – excellent – hit for a query. Therefore, the description of media objects has to be extensive – in order to cover all possible aspects – and the matching has to be very discriminative. Browsing is the mere opposite of matching. Descriptions may be short and feature transformation may focus on major aspects of the content. Brief descriptions do not allow highly discriminative categorization. Instead, this step will aim at a concise ordering of all media objects. Retrieval, eventually, is located somewhere between matching and browsing. Since more than one object has to be identified, the categorization cannot be as restrictive as in matching. In consequence, the descriptions need not be as long as in the matching case. On the other hand, retrieval can be seen as a very specific browsing task, i.e. the division of the feature space in media objects *relevant* or *irrelevant* to a particular query. Therefore, the categorization step has to be more discriminative than in the browsing case, and more feature information is required for a solid decision-making process. Table 10.1 summarizes these findings.

Application Type	*Feature Transformation*	*Description*	*Categorization*
Browsing	General	Short	Relaxed
Matching	Extensive	Long	Discriminative
Retrieval	Moderate	Medium	Moderate

Table 10.1: Requirements of Application Types.

Given the signals of the training set and the requirements of the application, the media understanding design process may flow as described in Figure 10.2. The first step is the analysis of the signal – as introduced in Chapter 2. The particular properties of the sample data (domain, bandwidth, etc.) in combination with the requirements determine the feature transformations that are selected. These feature transformations are then employed for the computation of descriptions, which are enhanced by information filtering methods. Refinement may already appear after this step (e.g. selection of different feature transformations), but usually potentially suitable categorization methods are selected and trained first. Methodological refinement is based on the evaluation results. The process of *method selection* for feature transformation, filtering and categorization is repeated and refined until the quality criteria are met. The eventual algorithm needs to be optimized in order to make it as fast as possible. Practically, this will not be done by the implementer of the media understanding

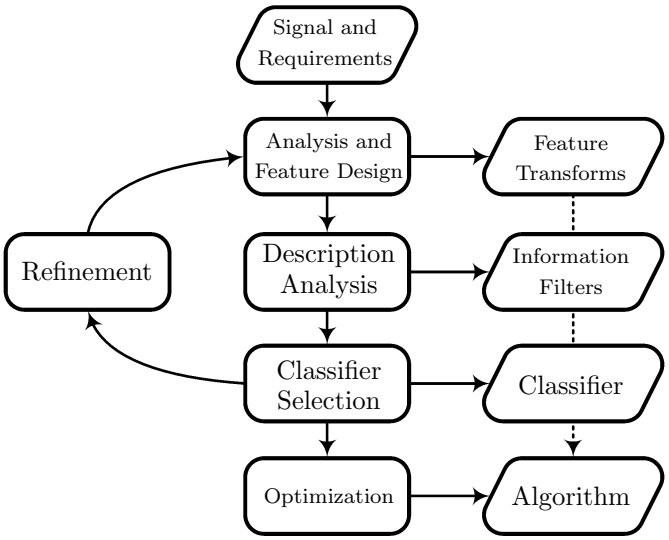


Figure 10.2: Media Understanding Application Design Process.

application. Rather, the applications will be designed in a flexible tool for media understanding while the eventual implementation will be performed using a low-level API and optimized libraries. In the next section, we will give examples for such programming environments.

The design process is straightforward engineering. It is requirements-driven in the sense that quality and needs have to be specified prior to method selection. Please refer to Chapter 25 for concrete examples of its usage. This chapter will introduce semantic features such as face recognition that necessarily imply an iterative media understanding process that adds categorization to the feature design loop.

The two major *high-level software design patterns* of media understanding are:

- A *processing chain* that implements the big picture of media understanding.
- An *iterative refinement process* based on *relevance feedback* that implements media understanding of media understanding.

The first point should be clear by now. Media understanding – whatever the media type is – is a process of summarization and categorization. The employed methods are signal processing (feature transformations), applied statistics (information filtering) and machine learning (categorization). The processing chain

will be a sequence of these methods – which, depends on the application domain, the application type and the state-of-the-art.

The second pattern has already been touched a couple of times in this book. Semantic understanding of media content can hardly be achieved with the low-level feature transformations of today (semantic gap). The major remedy against the semantic gap is applying an iterative process that puts the user and her semantic understanding (so-called relevance feedback) in the loop of refinement. Relevance feedback may be that some image is not relevant to some retrieval query or that some document is relevant to a query.

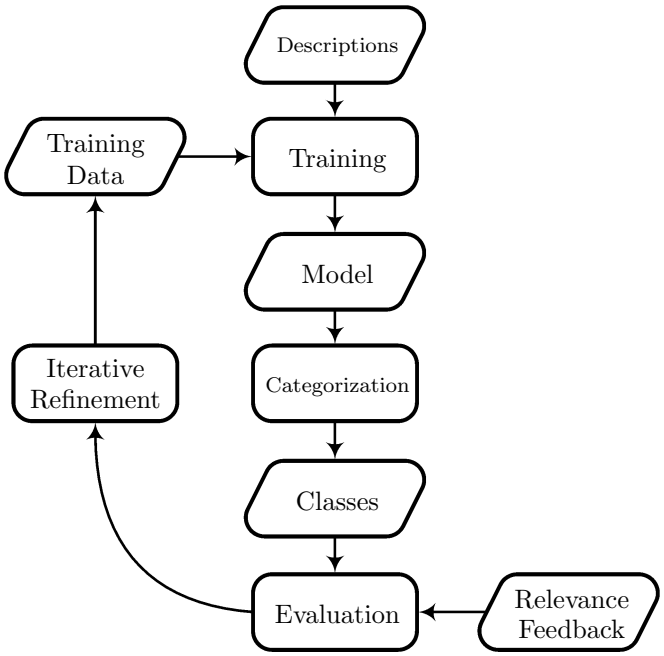


Figure 10.3: Iterative Refinement by Relevance Feedback.

Figure 10.3 illustrates the general iterative refinement process. The classes computed by some categorization process are evaluated based on relevance feedback. Then, some iterative refinement is applied to the training data, and the categorization process is performed again by firstly, training the model and secondly, applying it on the data. The iterative refinement step is responsible for computing the control information in this cybernetic process. Depending on the type of relevance feedback (human or ground truth-based) and the application domain iterative refinement may be performed in various ways. Two major forms are the following.

- *Ground truth manipulation*: In this case, a feedback of the form 'this item is relevant, this not' is employed to include/exclude some media objects from the ground truth with the effect that refined categorization models are trained. For example, if in a music retrieval application, all jazz samples are marked as relevant while all pop samples are marked as irrelevant only the first type will be included – as positive examples – in the training data of the classifier.

- *Re-weighting* : Here, the general idea is to manipulate the descriptions or probability distributions used for categorization by new – iteratively extracted – information on the problem domain. For example, the weight for the i-th description predicate (!) can be updated using the following formula:

$$w_i = log \frac{\frac{r_i}{R}}{\frac{n_i - r_i}{N - R}}$$

  Here, r_i is the number of relevant objects in the media database that have the i-th predicate on (for example, the i-th description element/term exists) while n_i is the overall number of media objects in the database where this term is on. Values R, N stand for the total numbers of (relevant) media objects in the database. This method is very popular in text understanding and a number of variations to the presented re-weighting scheme do exist.

Relevance feedback is of highest significance for successful media understanding. On the one hand, it has the potential to improve the results – and user satisfaction – dramatically while on the other hand the elimination of irrelevant components of the process may have a very positive effect on the performance of media understanding algorithms. Media understanding of media understanding may be seen as an elevator that increases the semantic level of operation with each iteration.

We would like to close the design section with a few remarks on the particulars of *mobile media understanding*. Such applications, for example, executed on a smartphone, have a high potential. Mobile speech recognition will soon be available at a quality that comes close to the Babelfish [2]. Visual applications will be usable for tracking and logging the environment. Many more interesting applications do exist. It is therefore just to discuss this specific scenario of media understanding here.

Generally, the design process is the same as for desktop applications. Figure 10.4 sketches this process (dotted elements). Only two components have to be added to the design process. The first is a *resource analysis*. In this step, the resources consumed by the processing chain have to be analyzed, and it has to be tested whether or not these resources are available in the mobile setup.

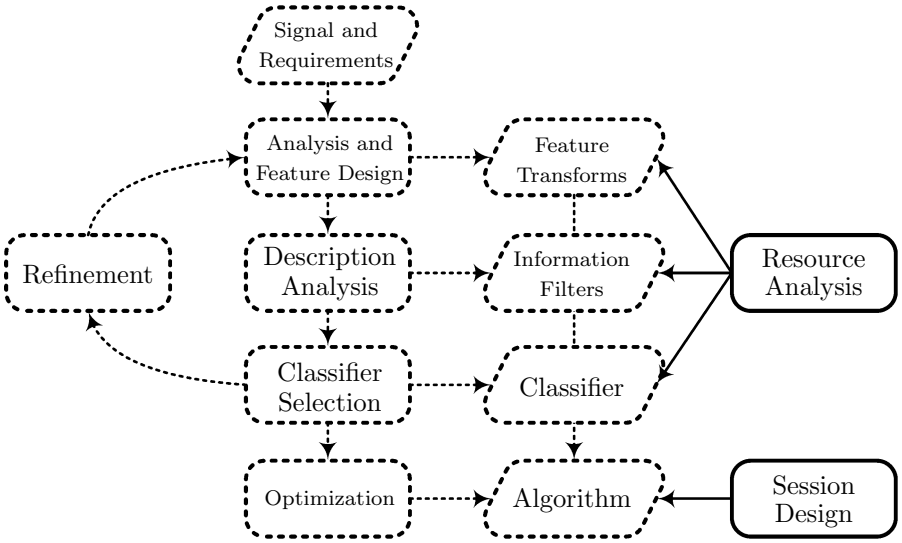


Figure 10.4: Design of Mobile Media Understanding Applications.

Practically, it can be very annoying to find out that the media understanding solution with acceptable quality is simply not doable in the mobile setup. In order to avoid such frustration it is important to perform the resource analysis before the implementation of the algorithm.

The second issue of mobile media understanding design is *session design*. That is, deciding which components of the applications have to run on the – limited – mobile device and which components can be outsourced to a remote machine with better processing capabilities. Generally, three strategies do exist:

1. Perform the entire application locally.

2. Perform only media capturing locally and everything else remotely.

3. Perform media capturing and categorization locally, but feature transformation, filtering and classifier training remotely.

In the first scenario, only very basic methods can be applied. The usage of the simple visual features described above limits the performance of a media understanding application on a state-of-the-art mobile device to less than one frame per second today. That is, of the 25 or more captured frames only one is available for feature transformation. Furthermore, only very simple categorization methods can be employed locally.

The second strategy is the most-employed today. For example, some Android media understanding applications follow the strategy to record only the reference media object locally and do everything else remotely. The advantage of this strategy is significantly higher processing power. The only disadvantage is the latency caused by media transfer over the network.

The third strategy may also not be neglected. Signal processing is certainly the most resource-consuming part of media understanding. It is, therefore, reasonable to source this part out while performing all other steps on the mobile device. In particular, if a well-trained Markov process is employed for categorization, such an application can produce satisfactory results at relatively low computation and networking costs.

In conclusion, designing a media understanding application means defining requirements, selecting appropriate feature transformations, filtering methods and classifiers, and implementing an iterative process for the optimization of each individual component and of the entire process. In the next section, we will discuss how this process is performed in practice.

10.2 Implementation

This section summarizes implementation aspects of media understanding. The implementation of such applications cannot be performed from scratch for every new problem. Instead, libraries for specific feature transformations and categorization algorithms do exist that can be reused and recombined. Below, we introduce the major sources of such libraries. Furthermore, we briefly discuss the problem of user interface design in media understanding as well as the practical provision of ground truth data and of probability distributions by sampling.

The fundamental problem of media understanding application programming is: When to use which feature transformation, information filter and classifier? The selection can be based on the design process outlined in the last section. More and more, however, a different scheme is employed in practice that consists of the following steps:

1. Apply all feature transformations on the media database that are available. In this step, the only limitation is the type of sample employed (quantitative or symbolic).

2. Apply information filtering methods on the descriptions in order to eliminate all elements that are redundant.

3. On the factors and some ground truth train all known categorization method and select the one for the application that shows the best performance.

The result is a media understanding application that employs all feature transformations that contribute significantly to the factors and the categorization method with optimal adaptation to the ground truth. Of course, this scheme relies strongly on a well-balanced, large ground truth. Otherwise, the result would be extreme overfitting to the training data. On the other hand, it is a domain-independent recipe that guarantees good results by simply taking into account everything that may help the cause. This quantitative scheme is the practical alternative to the human-centered design process suggested in the last section.

A number of tools and libraries do exist for the implementation of media understanding applications. In Appendix C we compare the following four major environments:

- *Matlab* [69]: A commercial software for sophisticated signal processing and machine learning (among others). Matlab provides *toolboxes* that contain all major feature transformations for the audiovisual domain, biosignal domain, etc. One of the major advantages of Matlab is its popularity in the scientific world. Hence, many toolboxes that were developed by scientific institutions are available free of charge and, often, provide the latest functionality.

- *OpenCV* [360]: This C-library (originally developed by Intel) provides functions for feature extraction from visual media. Furthermore, it offers the major categorization methods. Most algorithms are implemented very effectively, which makes the library tailor-made for the implementation of sophisticated visual media understanding applications.

- *R* [368]: R is a command line-oriented statistics toolbox that provides a variety of information filtering methods as well as classifiers. It also contains a programming language that allows for the implementation of new filtering methods. Similar to Matlab, R is widely used in the scientific world and, therefore, the latest algorithms are available for this system.

- *Weka* [378]: A Java-based application for the usage and comparison of categorization methods. Weka implements the vast majority of relevant categorization methods. Feature spaces with ground truth can be employed to train these categorization methods and compare their performance. Furthermore, Weka provides very useful information visualization methods.

In the appendix, we compare these four packages by their major features. It is highly recommended to employ R for any form of statistical evaluation and to use Weka for the identification of the best-performing categorization method. OpenCV is an excellent choice for feature design in the visual domain. Matlab

provides all kinds of functionalities and is, for example, state-of-the-art in the biosignal domain and the audio domain.[1] Major weakness of all packages is missing functionality for the processing of symbolic media data. In this case, we recommend the usage of a scripting language with powerful capabilities for regular expression parsing. The programming language *Perl* is unbeaten in this domain.

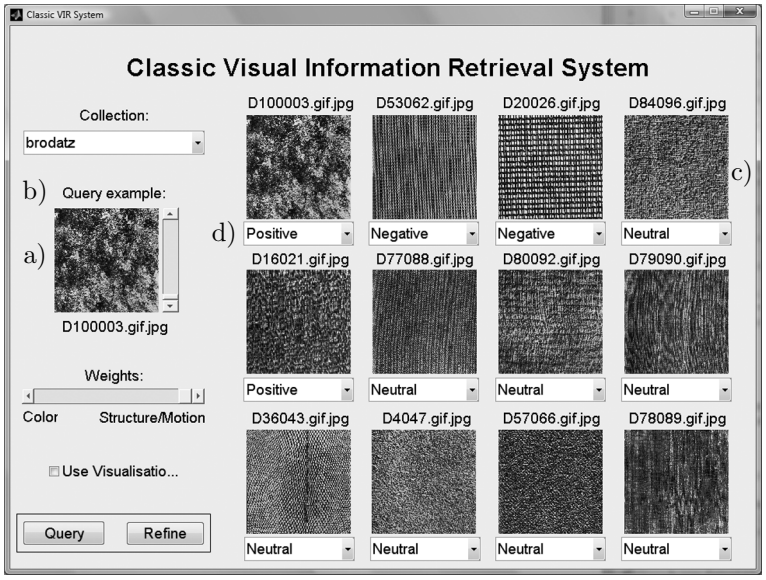


Figure 10.5: A Typical Retrieval User Interface.

Of the four packages listed above, only Matlab provides all functionalities required for application implementation, including a user interface design kit. User interfaces in media understanding require a few specific components – see Figure 10.5 for an example of a texture retrieval application:

- A component for the *media representation* (a). Visual media can be represented by their content (e.g. keyframes), text can be rendered symbolically, all other signals need to be represented by appropriate visualizations. Normally, the representation of audio by a graph of the waveform will not make sense while the representation by the name of the piece of audio will. The selection of a good media representation is often crucial for the understandability of the user interface.

[1] Links to more libraries, in particular, C- and Java-libraries for feature transformation can be found on the web page *atpress.info/mmir*.

- A *query definition* component (b) that allows the user to communicate his intention to the machine. Typical is the provision of one or more examples of the type of media of interest. In the figure, the query definition component is just a list widget that allows for the selection of one image.

- A component for the *result set* (c). A matching application requires just one widget for the match. For a retrieval system, a static grid of objects is sufficient. For browsing applications the result set needs to be dynamically adjustable since the sizes of clusters will vary.

- Eventually, a component for *query refinement* (d). In the figure, relevance feedback can be given on every image in the result set. Furthermore, the weights of the feature transformations employed can be adjusted by a slider. The selection of widgets for query refinement depends generally on the type of iterative refinement employed and the level of expertise of the users.

We would like to emphasize the importance of user-centered user interface design in media understanding. In the past, most of these applications were targeted at researchers and expert users. If media understanding should become an every-day tool for average users – like text-based search today – the user interfaces have to be easy to understand and straightforward to use. In the future, more effort should be laid on the user interface design aspects of media understanding.

One major source of data has to be provided for application building in media understanding: an as good as possible ground truth. Ground truth is appropriate to the problem domain if it contains examples for all likely cases and if the number of examples for of each type of description is related to the number of times such an event may occur in reality. It goes without saying that the construction of such a ground truth is a sophisticated, tedious undertaking. Moreover, it is an undertaking that – though of paramount importance – has little chance of earning the person that performs it scientific merit. It is, therefore, not surprising that only few ground truth libraries do exist that come near to the stated requirements. The better libraries are mostly not free of charge. The web page of this book lists a number of libraries that represent the state-of-the-art in their respective content domains.

Whenever a probabilistic method should be employed for categorization, the ground truth has to be transformed to distributions of conditional probabilities. In the last chapter, we described the major sampling methods – Gibbs sampling and expectation maximization – as well as the common approach to assume description elements independent of each other and hence, making the most of the available ground truth. Figure 10.6 illustrates two further heuristic approaches

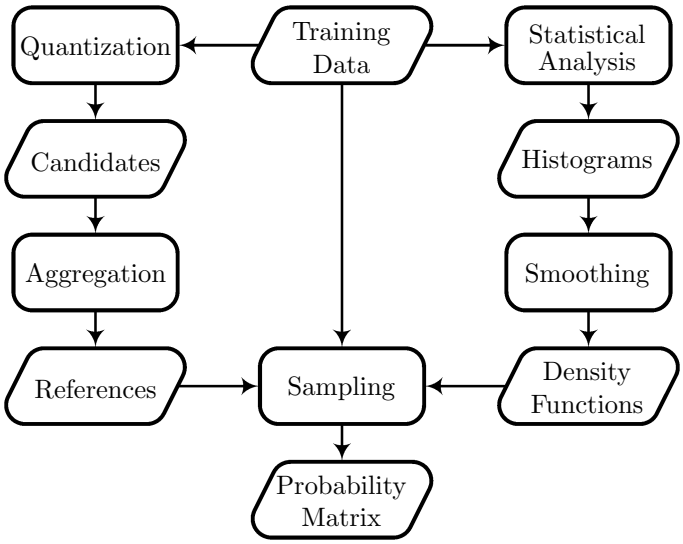


Figure 10.6: Improvement of Training Data for Sampling.

to overcome this problem. The center column of the figure stands for the approach described above. The left column is an approach for the construction of representative references as additional sampling sources. In order to do that, the available training samples are element-wise represented coarsely (quantization). Then, similar (for example, in terms of distance smaller than a threshold) candidates are aggregated using some statistical method (mean, weighted mean, median – all per description element). The result is a set of references that can be employed for density estimation.

The right column suggests another approach for practical sampling. In the first step, statistical analysis is performed in order to generate histograms of the descriptions in the ground truth data. These histograms are then smoothed and, thereby, transformed into density functions that can be employed as sources for sampling. This method and the two others depicted in the figure can provide valuable input to the sampling process.

In conclusion, the practical implementation of media understanding applications requires choosing an appropriate, powerful development kit. Appendix C compares the features of the major alternatives. Most media understanding applications are assembled from library functions. In order to implement the evolutionary prototyping approach outlined above, their performance needs to be evaluated – for example, by the methods discussed in the next section.

10.3 Evaluation

The first two sections of this chapter should have made clear that evaluation is the essential step in media understanding application building. The state-of-the-art evaluation process starts with the definition of *quality criteria* for the problem domain. For example, in a general-purpose face recognition application, the minimum quality level could be set at 95%, i.e. the application has to be optimized until at most 5% of all faces are associated with wrong identities. Formally, a quality criterion consists of a *measure* and a *threshold*. Below, we introduce the most important measures for ground truth-based evaluation. The thresholds depend on the application domain: 95% in face recognition is ambitious, but possible. Recognizing 99% number plates in a traffic surveillance application correctly may be doable while recognizing 60% violence scenes in feature films is hardly possible with state-of-the-art media understanding methods. Practical experience creates an understanding for what can be done.

In the second evaluation step, the prototype system will be tested against a pre-defined test set (usually, a subset of the ground truth), the measures will be computed for each run and, eventually, averaged over all test runs. If the measured values lie below the pre-defined thresholds the iterative refinement process is initialized in which methods and parameters are optimized until the requirements are met or until it becomes clear that they cannot be met by the available technology. This end is not too seldom reached in present media understanding.

If the quality criteria are met, a third evaluation step is executed: performance evaluation. In this step, the employed algorithms are optimized until the performance cannot be increased anymore without a significant loss in categorization quality. We would like to emphasize that due to the complexity of media understanding and due to the difficulties in reaching acceptable quality levels this third step is only of minor interest today. In the future, however, when computational media understanding will be comparable to human achievements in this field, it will become highly important.

The following list summarizes the steps of media understanding evaluation.

1. Define quality criteria
   (a) Select measures appropriate for the given test data
   (b) Define minimal thresholds of quality
2. Compute evaluation data
   (a) Take measurements for each run of the prototype system
   (b) Aggregate the measurements statistically

3. Repeat prototyping and the second step until the quality criteria are met

4. Optimize the algorithmic performance by employing heuristics, coarse representation, etc.

In summary, best practice in media understanding evaluation is a results-driven process, in which the algorithmic design follows clearly stated requirements. Below, we employ this scheme for ground truth-based evaluation. That is, we assume that a set of descriptions with associated class labels is available for evaluation. This situation may be considered optimal. Evaluation under less luxury circumstances is discussed in Chapter 20.

The following measures are typically used for evaluation if a ground truth is available. The measures assume a categorization situation with only two classes: relevant/irrelevant (retrieval) but, since every division into n classes can be reduced to a sequence of pair-wise comparisons, these measures are also applicable in browsing scenarios.

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad \textit{Recall} \tag{10.1}$$

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad \textit{Precision} \tag{10.2}$$

$$f = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \qquad \textit{Fallout} \tag{10.3}$$

Here, TP stands for the *true positives*, i.e. all media objects that were correctly categorized as relevant. The *false positives* FP were falsely categorized as relevant (first-order error), the *true negatives* TN were correctly categorized as irrelevant, while the *false negatives* FN were falsely categorized as irrelevant (second-order error).

Recall (also known as *true positives rate* or *sensitivity*) and precision are the two most important and practically used measures. The recall expresses how many relevant objects were really recognized while the precision expresses how relevant the returned set of media objects is. The fallout expresses the ability of the prototype system to sort out irrelevant objects.

Recall and precision are *interdependent* measures, since they are computed from the same components. One of the two measures alone cannot provide a full picture of a media understanding system. An experienced experimenter will be able to trade a better recall for worse precision (e.g. by very strict categorization) and the other way around. Therefore, a serious researcher will always evaluate her system by both measures.

In practice however, it is a non-trivial problem to provide a ground truth over a large problem domain. In such a situation, it is tempting to rely solely on the

precision values for evaluation, because the precision – in contrast to the recall – can be computed from the result set alone. The values for true positives and false positives can both be computed from the output of the classifier. Knowledge about the distribution of relevant and irrelevant objects in the entire database is not required.

It has, therefore, become common to describe media understanding systems that operate on large, diverse bodies of media objects by the precision alone, often by the so-called *prec@16* value. This value expresses the precision measured in a result set of 16 elements. The origin of this measure lies in the image retrieval domain, where $4x4$ grids are common result sets.

Common or not, precision alone is an unsatisfactory measurement. Recall and precision from a subset of the domain often express more about the capabilities of a media understanding system than the precision over the entire domain. Still, as we stressed above, recall can be traded for precision and vice versa. What we really want is a measure that rewards *strong recall and strong precision* while punishing every deviation towards neglecting one value. The measure that provides this behavior is the F_1 *score*:

$$f_1 = 2\frac{\mathrm{TP}^2}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} = 2\frac{r.p}{r+p} \tag{10.4}$$

The F_1 score is optimal, if recall and precision are balanced. If one value exceeds the other, the product grows slower than the sum, because any rectangle has a ratio of area and perimeter smaller than the – optimal – square. Figure 10.7 illustrates the relationship of F_1 score, recall and precision. If possible, the F_1 score should be employed for the evaluation of media understanding systems since it packs all relevant information into one value.

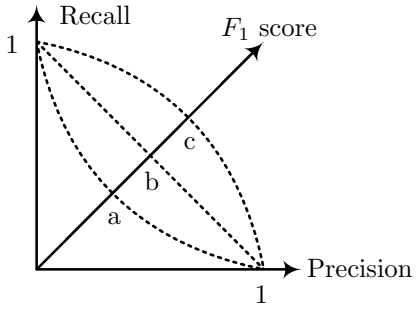


Figure 10.7: Performance Measures for Evaluation.

Figure 10.7 shows examples of *recall-precision graphs*. In such a graph, each point on the curve stands for one measurement of recall and precision in the

evaluation process. The entire curve expresses the performance of the prototype. In the figure, the dotted lines a, b, c express fundamental types of recall and precision relationships. Curve c is the desired behavior, rewarded by optimal F_1 score. Curve b is still acceptable, while curve a describes a system with inferior performance. In practice, the experimenter will endeavor to bulge line c as far towards the $r = p = 1$ point as possible.

Practically, the levels of recall and precision depend on the application domain and the size of the media database. For limited problems such as face recognition from a database with a few thousand nicely photographed pictures values of $r, p > 0.99$ are possible today. For other domains, such as general event recognition from a video corpus of one hundred hours or more, $r, p \sim 0.15$ is already a fair result. From this status quo, we can conclude, that there is still a lot to do in media understanding. However, if for some problem the quality is considered acceptable, performance optimization – as discussed in the next section – becomes relevant.

10.4 Optimization

Though overshadowed by the problem of quality optimization *algorithmic optimization* must not be neglected in media understanding. Most categorization methods introduced in the first part of this book employ micro processes that are comparatively complex (for example, compared to relational database queries). Even for small media databases, the answer time of a system employing such a classifier can easily become intolerable. Therefore, the practical usability of a media understanding application depends strongly on its performance.

Below, we briefly discuss two major areas of algorithmic optimization: heuristic query acceleration (supported by query prediction) and query acceleration by indexing. Generally, it has to be noted that *query execution* is in the focus of optimization. Feature extraction from the media database can be performed offline. The online query includes feature extraction from the query object (e.g. a reference) and categorization of the descriptions of the media objects based on the given reference. Hence, optimization of the feature transformations is of minor importance compared to optimizing the categorization process.

Heuristic query acceleration aims at shortening the execution time of a query by minimizing the number of times the micro process has to be applied. That is, the number of pair-wise similarity measurements is reduced as far as possible. The first step in this process is *query prediction*, i.e. estimating the time a query will require. Query prediction is typically implemented as a media understanding meta process. Experiences from past queries are analyzed using the instruments of time series summarization (for example, a sliding average) and extrapolation of the present query. The result is used to judge whether or not it makes sense

to employ a sophisticated query acceleration scheme or not.

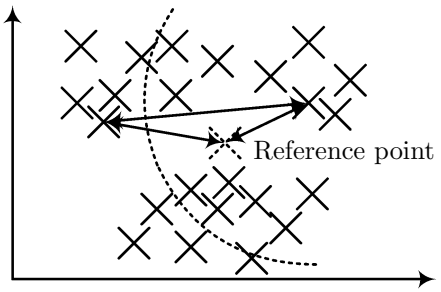


Figure 10.8: Query Acceleration based on the Triangle Equality.

One typical heuristic query acceleration method for distance-based categorization methods is to exclude media objects from the query process by usage of the triangle inequality (see Section 8.3). This method can be applied if only the first n objects should be returned (typically, retrieval). Figure 10.8 illustrates the approach. Offline, the distance of all objects x in the feature space to some reference point r (in the figure, dotted) is measured. During the query, the distance $m(x, y)$ of the query example y to any object in the database x needs only to be computed, if $m(x, r) + m(y, r) < m(x_n, y)$ where x_n is the description of the media object at the n-th position in the result set. That is, if the precomputed distances or their sum is already beyond the last object in the result set, the actual distance needs no longer be computed. Of course, this approach is only feasible for metric distance measures, i.e. if the triangle inequality holds. Its performance depends heavily on the chosen reference point. Generally, it is beneficial to compute the centroid of the feature space and use it as r. However, even under very good circumstances exploiting the triangle equality brings at most about 5-10% performance gain.

Indexing is a different approach to query acceleration that can be employed if the categorization micro process is based on pair-wise similarity measurement. The idea is to structure feature space in a way that allows to identify the subarea holding the likely answer to a query (in retrieval, a result set, in browsing, a cluster, in matching, the match). Trees are the natural data structures for this purpose. Starting from the root the subtree of relevance can be identified with a few comparisons of the reference to non-leaf nodes. The big disadvantage of this approach, however, is that normally, the reference object influences the categorization process. That is, *the morphology of the classes depends on the given references*. For example, the k-means algorithm (morphology determined by references) can only very ineffectively be accelerated by indexing while the k-nearest neighbor algorithm (reference defines only a local ad-hoc structure)

can nicely be represented by a tree structure.

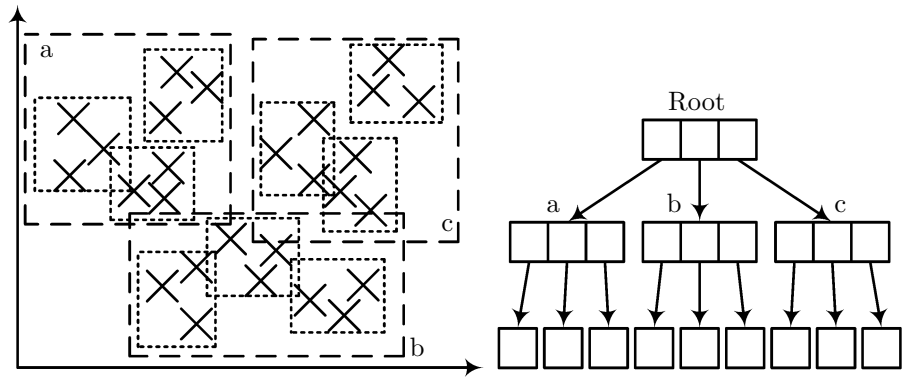


Figure 10.9: An R-Tree Example.

Practically, the *r-tree* and its several sub-forms [9] have proven very effective for the indexing of high-dimensional media descriptions. Figure 10.9 shows an example where a two-dimensional feature space is organized in triplets by minimum distance. The balanced r-tree is a derivate of the binary tree that uses similar algorithms for insert, delete and search. The major difference is that the r-tree allows areas (subtrees) to overlap. That is, one description may be part of more than one context. This fuzziness – similar to the one of decision trees – fits naturally to the requirements of media understanding.

However, the applicability of indexing in media understanding is very limited since the most powerful categorization methods – like Bayesian inference – employ disjunct concepts. Other starting points for general-purpose optimization include coarse representation of descriptions, compacting them by factor analysis, using the description elements with the highest influence on the factors instead of actually performing the factor analysis, and *grid computation*, i.e. performing the categorization micro process only for selected objects (references) and assuming all their neighbors having similar class labels. See [102] for more information on optimization.

Part II

Professional Media Understanding

Chapter 11

First Reflection and Bigger Picture

Summarizes the results of the first part, discusses the building blocks of feature transformations in detail, extends the big picture of media understanding and provides an overview over the second part of the book.

11.1 Conclusions from Fundamental Methods

In the first part of this book, we introduced a multitude of methods for feature extraction, information filtering and categorization. Before we continue with more advanced concepts of media understanding we consider it beneficial to review these findings, emphasize important points, identify communalities and equivalences and derive conclusions as a basis for the coming chapters. This chapter serves this purpose. In the first section, we emphasize the most important concepts of the first part. The second and third section are dedicated to the reflection of the methods introduced above. Section 11.2 analyzes and structures the feature transformation process. Chapter 21 will continue this analysis for the categorization process. Section 11.3 takes the results of the summarization process and refines the big picture. Eventually, the fourth section provides an overview over the methods discussed in the second part of this book. In summary, the present chapter summarizes fundamental media understanding and leads the way to advanced media understanding.

The remainder of this section is organized along the big picture. First, we list our major findings concerning the *media objects*, then for *feature transformation*,

information filtering and eventually, for the *categorization of descriptions*. The following list summarizes the paramount aspects of the digital media objects that form the basis of media understanding.

1. The overall goal is *understanding multimedia content* in the same way as humans understand sensual stimuli today. No doubt, media understanding science is still far from this goal. What is being done is primarily single-media understanding – if possible with an additional merging step in which the single-media results are joined. It is one purpose of this publication to establish links between the research disciplines focusing on individual media types. We have already seen above – and will see much more in this and the consecutive chapters – that the methods employed on data types as different as video and text are not that different. We cannot see why in the future fully integrated *multimedia understanding* systems should become implementable.

2. The *big picture* of media understanding defines the superior path through the media analysis and classification process. The big picture consists of a feature transformation step that summarizes large, redundant media chunks into well-defined media descriptions of fixed length. The descriptions are fed into a categorization process trained by a set of examples. The results of categorization are class labels that associate media objects with semantic categories. That is, the big picture suggests a summarization and specialization process that assesses general media content from a particular perspective (context).

3. *The media properties determine the employed methods.* Different types of media have different properties. For example, visual media are characterized by edges, i.e. sudden significant changes of pixel intensity. The human visual system focusses on such changes. Music, on the contrary, is characterized by the smooth composition of wave components. Human perception distinguishes pieces of music, among other properties, by their typical overtone structures. In consequence, audio and video media understanding require different methods and parameters. These differences are, however, not principally but often only the difference between using a sum operator on one type of media while employing the maximum operator on another. Later in this section we will discuss this topic in greater detail. In consequence, the media understanding researcher has to be aware of the specific properties of the individual media types.

4. In particular, the media types under consideration here (audio, bioinformation, biosignals, image, stocks, text and video) can be distinguished by their *fundamental types of samples*. Each digital media object is a composite of samples. Samples may be *quantitative* (measurements, e.g. audio,

image, video) or *symbolic* (elements of some set, e.g. bioinformation, text). Though there is a family resemblance between the two types and even hybrids do exist (for example, stock data), not all quantitative methods can be applied unadjusted on symbolic data and vice versa. It is important to understand the differences between quantitative and symbolic samples for successful multimedia understanding.

5. Eventually, we state a general morphological resemblance between media objects, descriptions and class labels. The sizes of these data types are fundamentally different due to the information filtering effect of media understanding. Their usage, though, is not. Media descriptions are still – abstracted – media content and class labels are judgments on media content with respect to a given context (query). In fact, in the media understanding process class labels are frequently used as additional – semantic – input for refinement processes.

The feature transformation step, often a signal processing application, constitutes the largest barrier between the single-media understanding disciplines. While most categorization methods are domain-independent (though sometimes used with different names and minor modifications), the feature transformations are usually considered very specific for the type of given media and understanding problem. Open-minded comparison of the methods reveals that, in reality, the differences are not that big. In the first part, we identified the following major aspects of the feature transformation process.

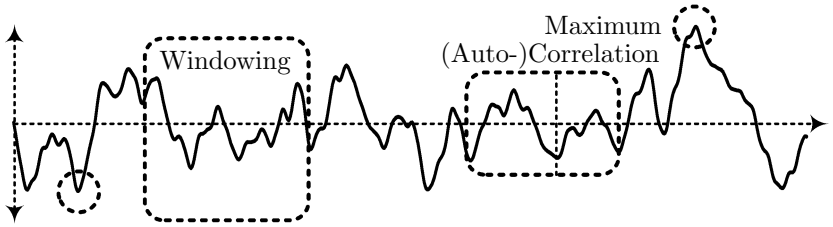


Figure 11.1: Summarization, Peak Detection and Correlation.

1. It appears that most feature transformation processes have a similar dramaturgy. Some localization step is followed by a weighting or convolution step over some template or a piece of the medium itself. Eventually, the results are aggregated again. We consider these elements the basic *building blocks* of feature transformation. Since the complexity of this issue is far beyond discussing it within the boundaries of this list, we dedicate the entire Section 11.3 to this purpose.

2. Besides their structural similarities, most feature transformations seem to pursue one of only three goals: description by *summarization* of media chunks (windowing), by *maximum* identification (peak detection), or by correlation of neighboring media chunks. See Figure 11.1 for audio examples. The first method performs reliable redundancy elimination. The second method summarizes by emphasizing important (in whatever sense) samples. The third method provides pattern recognition either within the media object (autocorrelation) or between a media object and some object of interest (crosscorrelation, e.g. with a face template). It is beneficial to keep these fundamental types of feature transformation in mind, because they allow for easy cross-media categorization of extraction algorithms.

3. Among the most important summarization-based feature transformations, we encountered the methods *zero crossings rate*, *short time energy*, *visual keywords*, *color histogram*, *statistical moments*, and *text summarization*. The zero crossings rate is as popular in audio understanding as it is in biosignal processing. Histogram methods are applied on almost all types of media while text summarization is, in fact, the same process on text as visual keywords are on visual data.

4. Outstanding maximum-based feature transformation methods are *attack time*, *dominant colors*, *edge extraction*, and *resistance/support lines*. The attack time measures an audio feature very similar to the (recurring) resistance line of stock data. Dominant colors are those with maximal visual impact on human perception. Edges are per se defined as points of maximal contrast.

5. Eventually, important correlation-based feature transformation methods are *linear predictive coding*, *MPEG-7 harmonicity*, the *correlogram*, *object contours*, *MPEG-7 color structure*, and *SAR textures*. Linear predictive coding is a typical autocorrelation transformation while object contours are normally based on template matching. SAR textures and color structures are hybrids, since they merge correlation with summarization and maximization aspects. However, the basic characteristics (identification of recurring patterns) are clearly of the correlation form.

6. Most successful feature transformations are influenced by *psychophysical findings* (see Chapter 23). Psychophysics describe the relationship between the outside world (reality) and subjective perception of this reality. Most of the fundamental methods are naïve in the sense that they do not explicitly address psychophysical issues. Instead, their popularity is a result of the coincidence that they fit with our perception. In the second part of this book, we will encounter feature transformations that exploit psychophysical knowledge explicitly.

7. The *MPEG-7 standard* was an important stimulus for the development of audiovisual feature transformations. In particular, in the audio part of the standard some very successful methods were defined for the first time. Unfortunately, audio and visual methods were developed separately ignoring the many similarities and common problems that would have existed for the two problem domains.

One feature extraction method is usually not sufficient to solve a media understanding problem of average complexity. Instead, a mix of transformation is typically employed. The result is the introduction of redundancy since any two methods will hardly produce orthogonal results. In this case information filtering has to be employed for redundancy elimination. So far, we identified the following major aspects of media understanding information filtering.

1. The generation of a *feature space by normalization and merging of descriptions* is a prerequisite of multimedia understanding. Normalization aligns the individual description elements along the same scale while merging joins descriptions originating from different transformations and/or media channels.

2. *Redundancy elimination* is the fundamental information filtering task in media understanding. Redundancy hinders the categorization process and has a negative effect on the algorithmic performance. Factor analysis, for example by principal component analysis, is a popular form of redundancy elimination but has the drawback that the resulting factors are hardly interpretable perceptually. However, this form of feature space optimization yields high-quality results.

3. The information filtering methods employed for summarization (statistical moments, regression, etc.) are related to both the summarization-based feature transformation methods (they summarize descriptions like the transformations summarize samples) and certain categorization methods. In particular, regression is the base of the risk minimization models introduced in Chapter 18.

In the categorization step of the big picture, we take the turn *from summarization to contextualization*. The two preceding steps filter the information content of the media source but do not alter it according to some context. Categorization is media interpretation. The resulting class labels are only valid in some given semantic setting. Hence, they express a high-level (perceptual) property of the media content that can be used in follow-up media understanding iterations as an additional type of descriptions. In the first part of the book, we have identified the following major aspects of the categorization process.

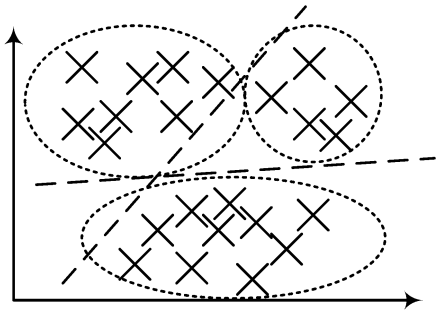


Figure 11.2: Categorization by Hedging (dotted) and by Separation (dashed).

1. The four fundamental approaches relevant in media understanding today are *rule-based*, *distance-based*, *probabilistic*, and *neural* categorization. Rule-based approaches discriminate by logical expressions and can be employed on any type of description, while distance-based approaches require quantitative samples and some form of underlying geometry (e.g. a vector space with interval-scaled dimensions). Probabilistic and neural categorization employ both complex weighting processes. In the first case, the weights come from general statistical facts about the media population under investigation. In the second case, the weights are derived from examples.

2. *Matching, retrieval and browsing* are the three fundamental purposes of categorization in media understanding. The 1:1 matching of media objects is a special case of retrieval, where a media database is split into a class of relevant and a class of irrelevant objects. Browsing requires associating a semantically valid class label to any member of the media population.

3. Almost all categorization methods can be split into a *macro process* and a *micro process*. The micro process performs the categorization of one or one pair of media objects against each other or a reference. The micro process is embedded in the macro process which encapsulates the entire process of *classifier training and application*. The need for training of the majority of methods implicates the provision of a *training set* and a *test set* of media objects. Usually, the training set is a subset of the test set.

4. Furthermore, categorization methods can be divided in two groups: *separators* and *hedgers*. Figure 11.2 illustrates the two groups. While separators try to put a division line between classes of media objects, hedgers try to rope in groups of similar objects. For example, the VSM classifier is a hedger that hedges around a query example and the k-means is a separator

that creates a Voronoi tessellation. Other hedgers are cluster analysis and k-NN. Decision trees and probabilistic methods are separators. Naturally, separators are tailor-made for retrieval tasks. Hedgers perform excellently in browsing tasks. However, both types of classifiers can be adapted to the requirements of both types of tasks with little effort (e.g. serialization of n separators for browsing). The distinction in separators and hedgers is only a preliminary step to understanding the communalities and differences of categorization algorithms. A more detailed analysis can be found in Chapter 21.

5. There is a fundamental difference between probabilistic categorization and the three other types. Probabilistic categorization can express *temporal relationships* between media objects. In particular, Markov processes provide this capability. In contrast, distance-based and other categorization models provide static judgments. Hence, these methods are not directly applicable in dynamic media understanding situations.

6. Eventually, there are *typical pairings of description types and categorization methods*. For example, histograms are often compared by distance-based models (for example, Minkowski distances) or by distance meta models such as the earth mover's distance. Shapes are frequently classified by meta models such as the Hausdorff distance. Words are typically compared by predicate-based measures such as the Hamming distance. These are just examples of successful combinations. In Chapter 21 we will provide a thorough analysis of this issue.

Most media understanding problems are far too complex for being solvable by one iteration of the big picture. The solution is *media understanding of media understanding*, i.e. the iterative application and sub-application of media understanding methods. Some problems can be overcome by simply repeating the media understanding process and applying *iterative refinement by relevance feedback*. In other cases, more sophisticated strategies are required. One example would be the usage of class labels as semantic description elements – thus feeding them back in the categorization process.

Furthermore, media understanding processes are often embedded in advanced feature transformation methods. For example, local visual feature transformations (see Chapter 14) usually employ a media understanding process inside the feature selection procedure. Of course, such processes, possibly combined with iterative refinement, are highly cybernetic and, therefore, require a sophisticated control mechanism.

Eventually, media understanding of media understanding can also be employed for performance optimization. Frequently, a rapid low-level procedure with simple descriptions, general filtering and fast categorization is employed

for pre-selection of potentially interesting media objects. Then, in internal refinement iterations, more sophisticated methods are employed for fine-tuning of the results. Such methods are typically applied on large-scale databases.

We would like to conclude this section with reminding the reader that despite all the clever methods introduced so far, overcoming the *semantic gap* is still a vary hard problem. For many real-world applications, more advanced approaches are required that will be discussed in the second part of this book. In the next section, we go into further detail on one fundamental component of the big picture of media understanding and structure the building blocks of feature transformations.

11.2 Building Blocks of Feature Transformations

We have seen that very similar – sometimes the same – methods are employed on different media types for feature extraction. Localization by windowing, for example, is employed on brain waves as well as video. What is different are mostly the media properties such as frequency or resolution. The fundamental algorithms are often the same. Below, we formalize this insight by defining a few *building blocks* that appear in most feature transformations. We then structure our knowledge about feature extraction along the raster of these building blocks.

The motivation for the introduction of building blocks is threefold. Firstly, a vague understanding of the methodological similarities of transformations is nice, but being able to structure arbitrary algorithms by the same components allows for detailed understanding of similarities and differences and, in consequence, for introducing a method in areas where it has been ignored so far.

Secondly, knowing and understanding the basic building blocks enables the practitioner to understand a newly introduced method easily and quickly. Media understanding research today is to a large degree about defining feature transformations tailor-made for a particular problem domain. Such transformations may become very complex. This hindering factor of understanding can be overcome if the methods can be structured into a few steps with well-known purposes.

Thirdly, from many years of media understanding research we have learnt that the one perfect solution to all media understanding problems is probably an illusion. Instead, narrowing down the problem domain and developing specialized solutions is a practically doable engineering scheme. For its execution, it would be highly convenient if feature transformations could be generated (semi-)automatically. Building blocks with well-defined interfaces could be recombined (for example, by a genetic algorithm) until a sufficiently good solution for given training data is found. This idea will be developed further after the introduction of the building blocks.

Figure 11.3 illustrates the building blocks of feature transformations and

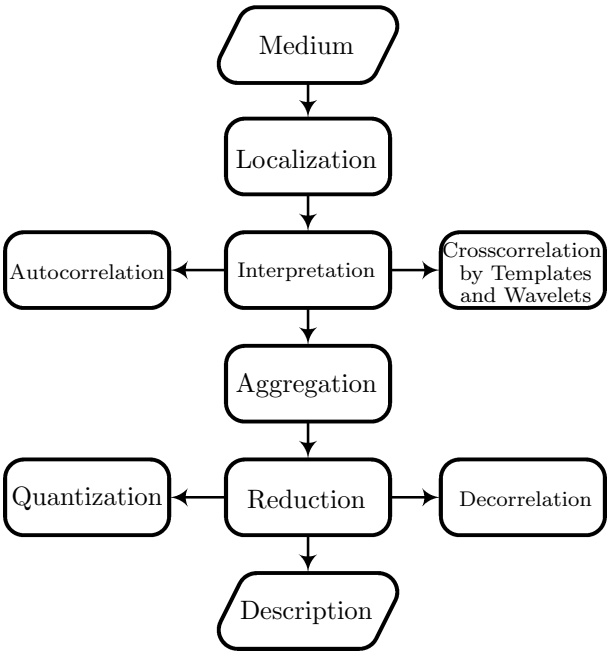


Figure 11.3: Feature Extraction Building Blocks.

their typical organization – the sequence may vary, of course. One fundamental element is a bracket of *localization* and *aggregation*. In Appendix A.6 we name these two instances of a $transform$ function loc_i and agg_i (i being the identification of one particular method). The localization function takes a media object and structures it in a set of *media chunks*. The localization may be performed along fixed window sizes, edges, contours, stop words or some other criterion. Important is, that one object is divided into many. For that, the neighborhood operator θ defined above can be used. The aggregation function performs the operation inverse to localization. Typical examples of localization and aggregation are the short time energy transformation which works on windows of fixed size and descriptions that are averaged over n windows. The computation of statistical moments is a common aggregation operation.

Within the localization bracket most feature transformations perform some *interpretation* operation, which can be of two fundamental types: *autocorrelation* or *crosscorrelation* (functions $acorr_i$ or $ccorr_i$, respectively). Both operations are based on convolution, which will be discussed in the next paragraph. In the first case, however, a media chunk is compared to a *neighboring* media chunk, i.e. from the same source, while in the latter case, a media chunk is compared to some

external object. A typical example of autocorrelation is linear predictive audio coding. Crosscorrelation is, for example, employed in edge detection (the Sobel operator is an external source), integral transforms (see Chapter 12), template-based face matching, etc.

It is very interesting to note that these building blocks which are of highest significance for all feature transformations that go beyond straightforward summarization are based on just one simple operator. *Convolution* is performed for autocorrelation as well as crosscorrelation of two media objects. The two cases can be written as follows:

$$acorr_i(o, \delta) = \theta(o, l, \cdot) \otimes \theta(o, l - \delta, \cdot) \tag{11.1}$$

$$ccorr_i(o, o_{template}) = o \otimes o_{template} \tag{11.2}$$

In the equations, we employ convolution based on the inner product, because this is the most frequent form. However, convolution based on difference may be employed as well (see Section 3.3). In the second case, some object o is convoluted over some template. In the first case, it is convoluted at all positions l with itself at distance δ. The template may, for example, be from the list in Appendix A.3, i.e. some wavelet, waveform, edge operator, etc.

First param	*Second param*	*Type of correlation*	*Example*
o_x	o_y	Global correlation	Object similarity
o_x	$o_{x-\delta}$	Spatial autocorrelation	Symmetries
o_t	$o_{t-\delta}$	Temporal autocorrelation	Linear prediction
o_x	$o_{template}$	Template matching	Face recognition
o_x	o_{normal}	Scaling	Gaussian blurring
o_x	o_{sine}	Spectrum computation	Fourier spectrum
o_x	o_{sobelh}	Edge extraction	Sobel edges
o_x	$E(o_x)$	Statistical moments	Mean

Table 11.1: Forms of Media Interpretation.

Table 11.1 lists a few examples of interpretation operations in feature transformations. The first two columns are the parameters of the convolution operator. In the last row, $E()$ stands for the expected value over all o_x. The table puts such important operations as scaling and spectrum computation in context with autocorrelation and template matching. Technically, indeed, all these operations are based on convolution.

Returning to Figure 11.3, the fourth major type of building block is *reduction*, which may occur as *quantization* or *decorrelation* ($quant_i$ or $dcorr_i$, respectively). Decorrelation covers all operations that eliminate redundancy (e.g.

factor analysis) while quantization covers everything from coarse representation to weighting (e.g. by psychophysical functions). That is, decorrelation considers the similarities *between* description elements while quantization focusses on the individual element. In this sense, reduction operations may be interpreted as filtering operations. Quantization need not necessarily be performed at the end of the feature transformation process. This step may also be executed before interpretation or even before localization or multiple times. For example, the adaptation of the feature extraction process based on relevance feedback may be performed before interpretation (e.g. weighting of media samples).

In summary, we propose four major building blocks of feature transformations: localization, interpretation, aggregation and reduction. Interpretation is always based on convolution while reduction may focus on isolated numbers or similarities between numbers. Table 11.2 provides a few examples of methods employed in the individual building blocks for different types of media. The localization and aggregation methods are very similar for the same types of samples (quantitative, symbolic). The interpretation methods are distinct for the types of media. In audio, time plays a very important role. Hence, temporal autocorrelation is a frequent tool. In the visual area templates are of paramount importance. Reduction, eventually, employs the same types of methods on all types of media but uses different *weights* – mostly derived from psychophysical findings.

Building block	*Audio, biosignals*	*Images, video*	*Symbolic data*
Localization	Windowing, band filter	Segmentation, scale space	Stop codons, punctation
Interpretation	Temporal autocorrelation, Fourier spectrum	Wavelet spectrum, template matching	Word count, motif identification
Aggregation	Mean, deviation, correlogram	Histogram, most frequent items	Bag of words, text summary
Reduction	Psychoacoustics	JPEG tables	Singular values

Table 11.2: Examples of Building Blocks per Media Type.

Before we conclude this section, we would like to employ the grid of building blocks for a discussion of the *properties of good feature transformations*. Generally, a well-performing feature transformation should fulfil the following requirements:

1. *Discrimination.* The resulting description elements should show high variance between objects that belong to different categories.

2. *Stability.* The description elements should have low variance for objects that belong to the same category.

3. *Performance.* The feature transform can be computed as quickly as possible.

4. *Efficiency.* The resulting description expresses the human categorization in as few values as possible.

5. *Generality.* The transformation employs as little as possible and as invariant/timeless as possible *context information.*

6. *Interpretability.* The description expresses a semantic, explainable category.

The first two requirements are the same as in canonical correlation analysis and linear discriminant analysis (see Chapters 16, 18 for details). It must be the paramount goal of a feature transformation to distinguish different classes by different values. The second pair of requirements aims at computation issues. The smaller the data the quicker the categorization can be performed. The third pair deals with context information. On the one hand, little context should be employed in the extraction process – for the sake of flexibility. On the other hand, the results should easily be interpretable. Obviously, these two goals – as the other pairs – are conflicting. It depends on the application designer to build a feature transformation that reaches the global optimum.

Requirement	*loc*	*acorr*	*ccorr*	*agg*	*quant*	*dcorr*
Discrimination		+	+	–		–
Stability		+	+	+		+
Performance	+	–	–	+	+	–
Efficiency	–	+	+	+	+	+
Generality		+	–		–	
Interpretability	–	+	+		+	–

Table 11.3: Influence of Building Blocks on Requirements of Feature Transformations.

Table 11.3 summarizes the influence of the building blocks on these requirements. Positive and negative influences are only given, where we see a clear relationship. Localization, for example, has a favorable influence on the performance, since it enables a divide and conquer strategy. Efficiency and interpretability are influenced negatively, because more values are generated and most localization methods do not separate chunks along semantic division lines.

Auto- and crosscorrelation are mostly positive for reaching the goals of feature transformation. Their major drawback is the bad performance. Crosscorrelation by templates, furthermore, reduces the level of generality. Aggregation, as a source of redundancy, has a negative influence on the discrimination capacity which causes a necessarily positive influence on stability. Performance and efficiency are both positively influenced by aggregating some numbers. Quantization and decorrelation both increase efficiency, the first one for the price of generality, the latter for less interpretability and bad performance.

We would like to conclude this section by pointing out how the building blocks of feature transformations can be employed for easier media understanding. Today, one of the biggest problems of media understanding is designing good feature transformations. Categorization, in comparison, is a much simpler problem since the best classifier for given test data can be selected automatically (e.g. by Weka [378]). Similar to that, feature transformations could be assembled from instances of building blocks (classes of methods) with standardized interfaces. Localization, aggregation, interpretation and reduction could easily be described by input and output functions (e.g. $1 : n$, $m : n$). Standardized methods given, identifying the best feature transformation and classifier for given test data would be reduced to a recombination problem. One solution could be the description of the process by a *gene string* and optimization using a genetic algorithm with a media understanding evaluation function for assessment. In consequence, media understanding research would be reduced to the definition of new instances of building blocks, not their (tedious) recombination.

11.3 A Bigger Picture of Media Understanding

In this section, we distill the results of the two previous sections and augment the big picture of media understanding. After introducing it we discuss context and semantics, refinement based on evaluation and ground truth and the resulting cyclic media understanding process.

Figure 11.4 shows the bigger picture of media understanding. The dotted elements extend the original picture by components that were discussed in the first part. Feature extraction is influenced by additional sources of information subsumed as *context*. The same data are also fed into the categorization process – in particular, during training. Training of classifiers is, furthermore, controlled by the *training data* in the form of ground truth and/or references. The resulting categorization is *evaluated* by the methods discussed in Chapter 10 and forwarded to a refinement procedure that manipulates the feature extraction process as well as categorization. The cycle of feature extraction, categorization, evaluation, refinement is another instance of *media understanding of media understanding*. If the evaluation employs user input (*relevance feedback*) we call

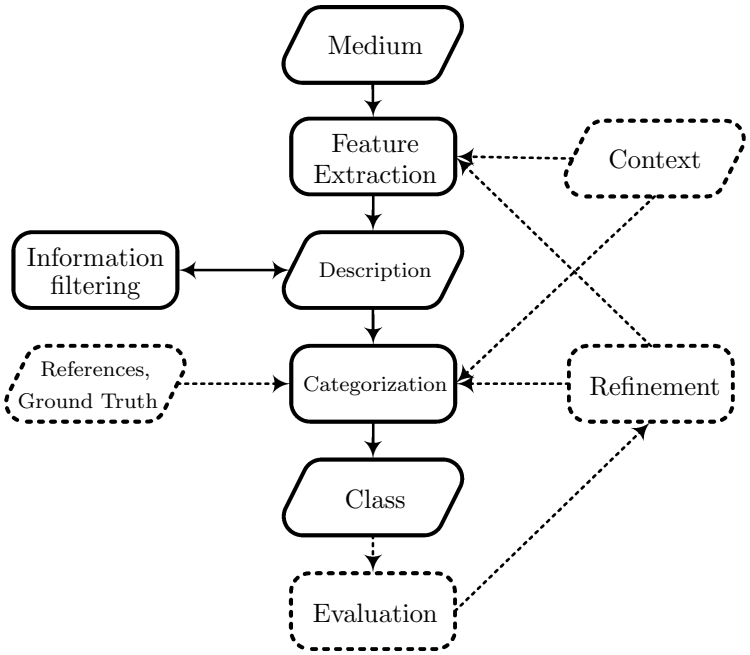


Figure 11.4: Bigger Picture of Media Understanding.

this *putting the human in the loop.*

Context influences the two major steps of media understanding. It must, therefore, be something important. Also known as *world information* context comprises all information relevant to the problem domain. Figure 11.5 names a few types. The inner circle lists more typical forms of context, the outer specific and technical instances of context. Location and time are arguably the most important types of context. The time when something happened, and the location where it happened have a definite influence on the semantic setting of a (media) event. Quality of service and battery status are important forms of context in mobile setups. These aspects may influence the media understanding process as a whole by, for example, selection of battery-saving extraction methods. Other types of context, such as topic and line of business are only relevant for specific problem domains. The figure lists just a few types of context. Many more do exist.

The application of context reduces the level of generality of a media understanding solution. It should, however, have a positive influence on both discrimination ability and interpretability by accelerating the specialization process that is performed in the categorization step. Using context in media understanding

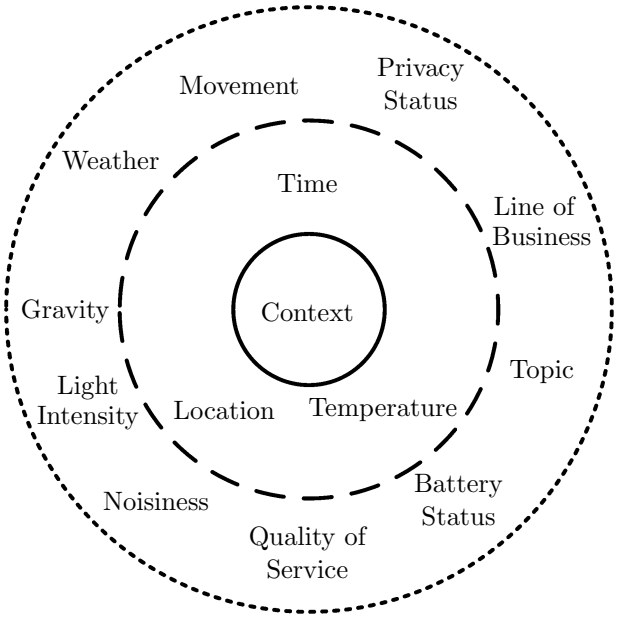


Figure 11.5: Some Types of Context Information.

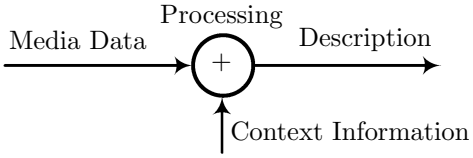


Figure 11.6: Application of Context Information.

applications will always be performed with the goal to increase the semantic level of the resulting categorization. Technically, the application can be performed as described in Figure 11.6. There, context data is employed to *control* the processing of the media data. A typical example would be the usage of spatiotemporal context for the elimination of irrelevant media samples, i.e. rule-based filtering like in a decision tree. Another option would be the filtering and/or adaptation of ground truth/references. Sometimes though, context may be source as important as the media data. For example, information about varying temperature can be an independent data stream in environment surveillance. Simply, spatiotemporal context data can be treated as additional description elements.

The second element that extends the big picture is the training data for

the categorization – present in the form of references or ground truth. Such data are crucial for the specialization process. They may, therefore, be seen as a very particular type of context information: *use cases*. In fact, training data are more relevant than the actual categorization method since the latter is typically selected based on its performance on the training data. In consequence, one of the most important steps in solving a media understanding problem is assembling high-quality training data. Such data must fulfil the following two requirements:

1. It must contain examples for all cases that may appear practically.
2. Ideally, all cases should be represented with the same frequency with which they occur in reality.

Both requirements appear trivial but are – for practically relevant problem domains – very hard to fulfil. In fact, assembling an expressive ground truth is more an art than science. Considering further the (unjustly) low potential for scientific merit and the high costs explains why assembling high-quality training data is one of the most neglected issues in media understanding today.

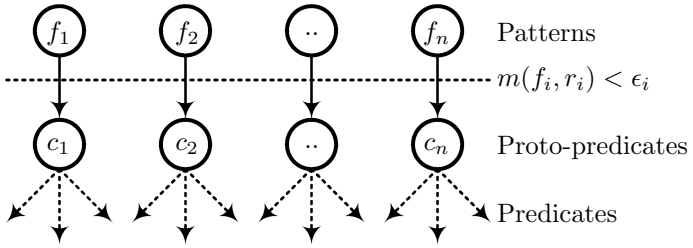


Figure 11.7: From Patterns to Predicates.

Evaluation – either based on training data or user feedback – leads to refinement of the media understanding process. Iterating this loop is one way to narrow the semantic gap between low-level methods and high-level demands. One particularly important cycle of iterative media understanding is illustrated in Figure 11.7: the transition from *quantitative descriptions* to *qualitative descriptions* (class labels, predicates). If the media samples are quantitative, this process transforms measurements into predicates that describe the media data from the perspective of training data, context and categorization method. Naturally, the resulting predicates can be employed as another source of context and fed back into the feature extraction and/or categorization process. Technically, the transgression from quantities to *proto-predicates* can be based on any categorization method. Frequently used approaches are rule-based and distance-based

methods. In the figure, for example, we measure the distance of some sample f_i to a reference r_i (given) by some measure m. If the distance is smaller than some threshold ϵ_i we consider a particular property given ($c_i = 1$) otherwise not. For example, if a particular stock value f_i is above a given resistance threshold, we consider the market under consideration to be booming.

In conclusion, the bigger picture of media understanding contains all major components of a professional media understanding system – no matter if focused on audio, visual or symbolic material. We have emphasized that, eventually, the quality of the ground truth is decisive for the quality of the media understanding process. Providing high-quality ground truth information is a non-trivial time-consuming task but most likely worth the effort. Furthermore, provision and application of context information are of highest importance for the bigger picture. Since the bigger picture is a realistic carcass of the media understanding process, we will structure the second and third part along its flow of information. The last section of this chapter provides a brief overview over the second part.

11.4 Overview Over Advanced Methods

Like in the first part, the chapters of the second part are organized along the big picture. That is, first we deal with advanced feature transformations, then advanced filtering, advanced categorization and, eventually, advanced evaluation. In feature transformation, we have two foci. Firstly, the computation and usage of *spectral descriptions* and, secondly, computation and usage of *local descriptions*. The focus in categorization is on *risk minimization*. Furthermore, the *dynamic categorization* methods already introduced in the first part will be extended by sophisticated, practically highly relevant methods. The remaining paragraphs of this section give a brief outlook on the contents of the second part.

Above we introduced localization as a base function of feature extraction. By spectral transforms, we intend to free our features from the drawbacks of the temporal domain – for example, the inflexibility of neighborhoods of samples. We introduce *unitary transforms* as the means for the averaging of spectrally related components. Digital transforms are one application of crosscorrelation by waveforms and wavelets. The result of integral (digitally, in fact, summarizing) transforms is a *spectrum*. Many advanced feature transformations in the audiovisual domain are based on spectra. Good knowledge of such methods is of fundamental importance for the media understanding expert. Hence, we discuss unitary transforms and spectral feature transformations in two chapters.

Localization is the second focus of feature transformation. We already mentioned that the human eye is a scanner that generates a stream of local descriptions. In the visual domain, local features imitate this behavior. However, localized methods are equally important in the other domains. We will discuss

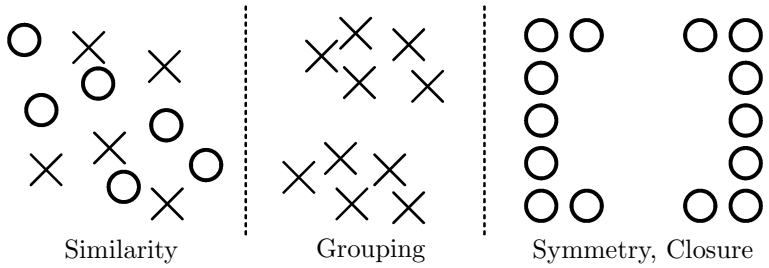


Figure 11.8: Examples of Gestalt Laws.

methods for the detection of *interest points*, i.e. neighborhoods with significant content, as well as their description. Solving the latter problem requires achieving invariance against rotation and other transformations. We will, furthermore, make a first step into local semantic features by discussing Gestalt properties (Figure 11.8) and their influence on interest point selection. Eventually, *description of motion* will be discussed as an application of local media features.

The advanced information filtering chapter will extend the fundamental methods discussed in the first part. We will discuss feature selection, merging of variable-sized descriptions and important signal processing operations such as source separation. The goal of this chapter is to make the reader familiar with a set of practically relevant tools without going too deep into the theory of information filtering.

The categorization chapters of the second part start with a formalization of machine learning. The methods discussed in the first part will be set into context, and gaps (mostly of theoretical nature) will be filled with the current state of the theory. We will discuss the differentiation of categorization methods into hedgers and separators but as well show that for some problems of machine learning, no satisfactory solution has been identified yet.

The introductory chapter prepares the ground for sophisticated categorization methods that learn over time. One chapter discusses all methods based on the *risk minimization* principle. These methods are mostly based on the linear regression model. However, by the introduction of some very nice twists new, extraordinarily powerful classifiers have been developed. We will endeavor to explain how they work and what similarities exist between them. These methods are typically separators.

The dynamic categorization models discussed in the subsequent chapter are mostly based on the principles of probabilistic categorization. For example, *expectation maximization* (introduced as one solution of the sampling problem) will be employed for the learning of classifiers that base their decisions on a reasoning somewhere between beliefs and risk minimization. Naturally, such

methods are hedgers.

Eventually, the evaluation chapter will mop up everything that could be said about the evaluation topic but was not in the first part of the book. In particular, we will discuss what can be done if no ground truth is available. The methods will be set into context and novel solutions for yet unsolved problems will be introduced.

In summary, the second part introduces a vast number of practically relevant media understanding methods. It is based on the material presented in the first part. We intend to provide everything in the second part, that is required for the practical application of media understanding. The third part will go beyond this state by bringing up problems of active media understanding research.

Chapter 12

Transforms in Media Understanding

Introduces the concept of spectral representation by discrete transforms, discusses several continuous and time-limited bases, distinguishes wavelet functions by their characteristics and introduces two major parametric transforms.

12.1 Introduction to Unitary Transforms

In this chapter we provide the basis for *spectral descriptions* – a highly important group of feature transformations for quantitative media sources. Spectral descriptions are generated from the media source by *discrete unitary transforms*. This section introduces the general model of discrete transforms, its first major exponent, the Laplace transform and discusses applications relevant to the media understanding domain.

The motivation for using discrete transforms is that they compute a *spectrum*. For our purpose we see the spectrum of some signal as the *set of coefficients* (weights) of some – to be defined – *base functions* with the property that the sum of weighted base functions is equivalent to the signal. The base functions are well-known and fixed. Hence, the coefficients are representative for the signal. Now, if we choose the base functions wisely, we can gain significant advantages such as simpler computation and easier interpretation of the spectrum than of the original signal. For example, if the base functions are sine waves, the music coefficients can easily be interpreted as the overtone structure. In the first part we introduced the *gravity of the sample* problem (semantic gap) as the difficulty

to see the semantic meaning in assemblies of samples. Discrete transforms help to overcome this problem, since in the spectrum each coefficient aggregates over *all* samples. Generally, discrete transforms help interpretability by transforming a time-based signal to one that is based on a – hopefully, semantic – set of base functions.

The general model of spectral transformation is the *integral transform* of the following form:

$$o_s(y) = \int_{-\infty}^{\infty} k(x, y)o(x)\,\mathrm{d}x \tag{12.1}$$

Here, o is the media source, o_s is the spectral representation, x is the iterator for the time domain and y is the iterator over the base functions $k(x, y)$. The general model of the integral transform was developed from the Laplace transform. It establishes an isomorphism between the time-based signal and the spectrum based on functions $k(x, y)$. The back transformation is defined as follows:

$$o(x) = \int_{-\infty}^{\infty} k^{-1}(x, y)o_s(y)\,\mathrm{d}y \tag{12.2}$$

That is, transform and back transform are symmetric. There is no loss of information. The transform provides just a base change. Often, $k(x, y) = k^{-1}(x, y)$. This equivalence is, for example, true for the Laplace transform and the Fourier transform (see below).

From the definition it becomes clear that integral transforms are not applicable to symbolic data sources. The entire model assumes some degree of *neighborhood* between samples. Since the base functions are typically smooth, neighboring samples are aggregated with similar weights. Such a concept of neighborhood does per definition not exist in symbolic media. Hence, spectral transformation is only relevant to quantitative media domains.

The general model of integral transforms is nice but not directly applicable to digital media objects. In the discrete domain, Equation 12.2 is written as follows:

$$o_s(y) = \langle k(x, y), o(x)\rangle = k(x, y) \otimes o(x) \tag{12.3}$$

A discrete unitary transform is nothing else than convolution of some signal $o(x)$ over some base $k(x, y)$. Of course, the back transform is defined equivalently:

$$o(x) = \langle k^{-1}(x, y), o_s(y)\rangle = k^{-1}(x, y) \otimes o_s(y) \tag{12.4}$$

The set of base functions $k(x, y)$ may be seen as a *template* representative for the media domain. For example, for audio signals a set of sine waves may be

suitable. For the visual domains (mind the importance of edges!) a set of step functions may be suitable, etc. We will discuss this issue – reasonable pairings of media sources and base functions – in the next two subsections. The convolution of source and template is maximal if the two signals are identical. Hence, *the resulting spectrum measures the degree of similarity between signal and base.*

Since all practically relevant discrete transforms are based on positive convolution, this type of operation is an *interpretation* of the media content, specifically a *crosscorrelation* of media data and base functions. It is general practice to employ the positive convolution for spectrum computation. However, negative convolution could be employed as well. Then, the spectrum would measure the difference between signal and base functions. As we mentioned in the first part (and will discuss further in the third), positive convolution is a similarity measure for separable stimuli (e.g. lists of predicates) while negative convolution is a distance measure for integral stimuli (e.g. lengths of objects, but as well wavelengths). The quantitative media objects subject to discrete transform are integral by nature. Hence, negative convolution may compute spectra that can more easily be interpreted. Unfortunately, hardly any research has been conducted on this question so far. It would be interesting to see whether a transformation based on negative convolution would be superior for some types of media.

The usage of discrete transforms in media understanding may appear arbitrary without the knowledge of the origin of these transformations, their typical applications and their interpretation. The following list provides the history in brief form.

1. Laplace developed his transform in 1785 with the intention of being able to solve certain differential equations more easily in the spectral domain. Inspired by the works of Euler, he established the entire model and introduced the base $k(x, y) = e^{-xy} = k^{-1}(x, y)$. The Laplace transform allows to represent integration and differentiation in the time domain by multiplication and division in the spectral domain. However, the existence of a back transformation was of highest importance, since the result was needed in the time domain.

2. In 1822, Fourier extended Laplace's base to $k(x, y) = e^{-ixy}$ which may appear weird at first sight. However, according to Euler's formula $e^{-ixy} = \cos(xy) - i\sin(xy)$ (see next section), i.e. the complex spectrum can nicely be interpreted as a set of sine waves with different frequencies. The back transform is equivalent to the forward transform.

3. In 1909, Haar introduced the first known set of base functions that was not *continuous* but *time-limited*. This *Haar wavelet* will be discussed in Section 12.3. Time-limited base functions can more easily be applied on

discrete data sources than continuous – an advantage that hardly mattered to Haar (who was still interested in solving differential equations) – but which became very important when computational image analysis started to develop in the 1950ies.

4. Around 1950, Gabor tried to overcome the problems of continuous base functions by a windowed Gaussian function (see Section 12.3). The wavelet idea was yet undiscovered then, so, paradoxically, the Gabor function is an intermediate step between continuous and time-limited bases that was developed decades after the satisfactory final solution.

5. Around 1990, the wavelet theory was fully developed, the Haar wavelet was rediscovered and others were introduced. Wavelets became, for example, state-of-the-art in image compression and were also used in image understanding.

This brief history should make clear that discrete transforms where developed for applications completely different than media understanding. For example, in media understanding *the existence of a back transformation is irrelevant.* The spectral description has to be representative of the media object but it is not required to reconstruct the signal from the description. Furthermore, we only consider digital signals that are somehow windowed. Therefore, continuous base functions are only of limited interest in media understanding. In summary, for media understanding base functions can be viewed best as templates and discrete transforms as the act of interpretation that allows to overcome the gravity of the sample using the template as a semantic bridge.

Not all transforms were designed as unitary. Two major exceptions are the Radon transform (1917) and the Hough transform (1959) which are more or less equivalent. Transforms that do not have a back transform are frequently called *parametric transforms* since the output depends on the control parameters of the transform – for which unitary transforms would provide no space.

The remainder of this chapter is structured in three sections. In the next, we introduce the Fourier and the cosine transform, two examples for transforms that employ continuous bases. Section 12.3 discusses wavelet transforms, i.e. transforms that employ time-limited bases. The last section discusses the Radon and Hough transforms as two parametric transforms that are very important in for media understanding.

In conclusion, the introduction of spectral transforms opens an entirely new domain for feature extraction. Chapter 13 is dedicated to spectral descriptions. The Laplace transform is the role model for all unitary transforms. However, in media understanding it is only of theoretical relevance. Practically relevant are the Fourier and cosine transforms discussed in the next section.

12.2 Transforms with Continuous Bases

This section deals with discrete transforms that employ waveforms as bases, i.e. signals that are defined for the interval $]-\infty,\infty[$. Two transforms are of outstanding importance in this domain: the *Fourier transform* and the *cosine transform*. We discuss their communalities and differences based on the spectra created for visual and audible examples.

The Fourier transform (FT) is employed in the same way as the Laplace transform, only the base function is different:

$$o_s(y) = \langle k(x,y), o(x) \rangle = k(x,y) \otimes o(x) \tag{12.5}$$

$$k(x,y) = e^{-ixy} \tag{12.6}$$

The back transform – though hardly relevant for media understanding – is symmetric to the transform.

As pointed out above, the *Fourier kernel* may appear surprisingly complex at first. However, it is equivalent to a complex combination of sine waves.

$$e^{-ixy} = \cos(xy) - i\sin(xy) \tag{12.7}$$

This equation is known as *Euler's formula*. It can easily be proven using the Taylor series expansions of e^x and the trigonometric functions.

$$\begin{aligned}
\sin(x) &= \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} + ... \\
\Rightarrow -i\sin(x) &= -i\frac{x}{1!} + i\frac{x^3}{3!} - i\frac{x^5}{5!} + ... \\
\cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + ... \\
\Rightarrow \cos(xy) - i\sin(xy) &= 1 - i\frac{x}{1!} - \frac{x^2}{2!} + i\frac{x^3}{3!} + \frac{x^4}{4!} - i\frac{x^5}{5!} + ... \qquad (12.8) \\
e^x &= 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + ... \\
\Rightarrow e^{-ix} &= 1 - i\frac{x}{1!} - \frac{x^2}{2!} + i\frac{x^3}{3!} + \frac{x^4}{4!} - i\frac{x^5}{5!} + ... \qquad (12.9)
\end{aligned}$$

Equations 12.8 and 12.9 are obviously equivalent.[1]

Hence, the FT has a major advantage over the Laplace transform. The (complex) coefficients can intuitively be interpreted as weights of sine waves with

[1] Euler's formula is also the basis of Euler's identity $e^{i\pi} + 1 = 0$. No relevance for media understanding, just beautiful.

increasing frequency. This property makes the Fourier transform very interesting for some areas of media understanding, namely audio understanding, biosignal understanding and stock analysis. The only problem is that the Fourier spectrum is complex. In practice, we often use the real part or the complex part of the Fourier transform only. Then, we speak of the *real FT* or *cosine FT*, etc. Of course, the back transformation is lost in this case. The attempt would result in a phase shift.

The FT has a number of mathematical properties that make calculations in the spectrum easier than in the time domain. For example, linear combinations in time domain remain linear combinations in the frequency domain (spectrum). Shifts in time domain or frequency domain cause weighting by some $e^{f(x)}$ in the other domain, etc. Some of these properties cause nice side-effects in media understanding that will be discussed where encountered. Before we investigate the Fourier spectrum a bit closer, one further term has to be mentioned. The *fast Fourier transform* is an algorithm that reduces the complexity of computation from $O(n^2)$ to $O(n \log n)$. As frequently the case in dynamic programming (see Chapter 19), this reduction is reached by replacing one loop by a recursion – thus shifting part of the algorithmic complexity to the local memory.

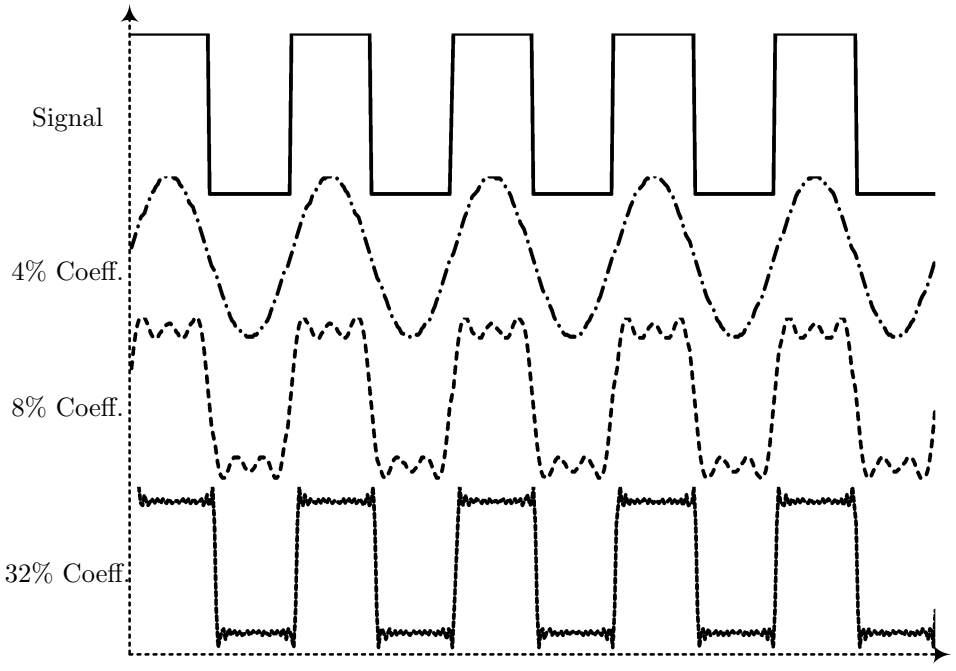


Figure 12.1: Fourier Approximation of a Square Wave.

Figure 12.1 shows three different Fourier spectra for a square wave (400 samples). The first line shows the signal. The second to fourth lines show spectra with 16, 32, 128 coefficients (4, 8, 32 per cent of samples), respectively. That is, all higher coefficients computed by the discrete transform were omitted. The result is a *band-limited signal* with the given number of coefficients. As can be seen from the graphs, the approximation becomes more accurate for less limited spectra. The first approximation is a pure sine wave. Doubling the number of coefficients adds overtones that improve the approximation. At 32% the approximation is already fair.

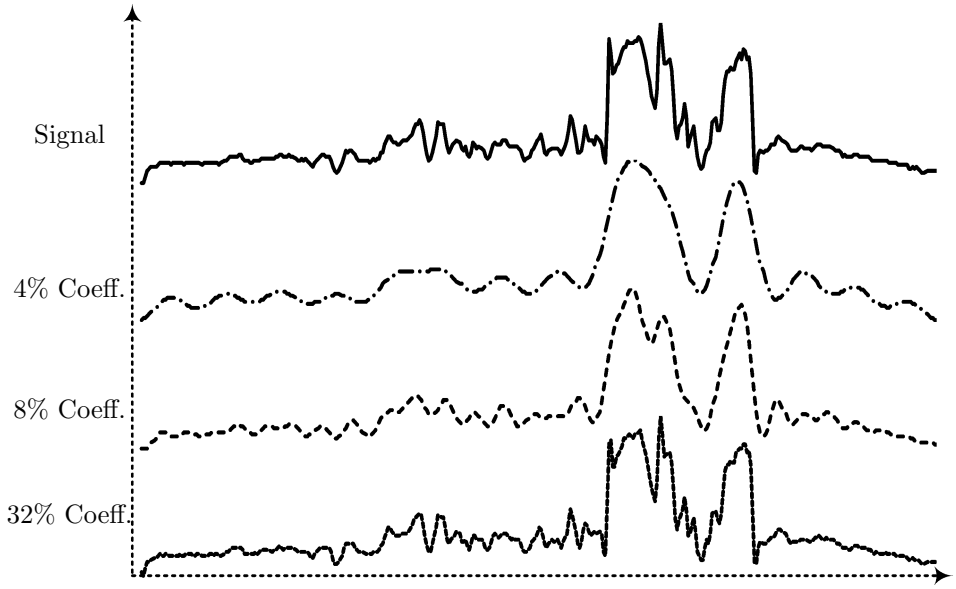


Figure 12.2: Fourier Approximation of an Image Signal.

Figure 12.2 shows the same process for a typical image signal (the center line of the leading example shown in the next figure). As can be seen, the approximation at 4% coefficients is already surprisingly good. Major difficulties arise at the peak section at approximately two thirds of the signal (presumably, edges). This result is typical for the Fourier transform. The first approximation is good for smooth signals but bad where sudden changes occur. Adding coefficients improves the reconstruction linearly. That is, *the information of the signal is distributed over all coefficients of the Fourier spectrum.* This property makes it tailor-made for audio understanding where such a spectrum provides valuable information on rhythm patterns and overtone structures.

For image information and human visual perception of edges, the Fourier

transform is not ideal. The *discrete cosine transform* (DCT) outperforms FT on such signals. There is no general definition of the CT. Instead, several variations do exist. One typical definition is the so-called DCT-II that uses the following base signal.

$$k(x,y) = \cos\left(\frac{\pi}{n}\left(x+\frac{1}{2}\right)y\right) \tag{12.10}$$

Before we continue with comparing Fourier transform and cosine transform visually we have to explain that both transforms (in fact, most discrete transforms) are *separable*. That is, the same algorithm can be applied on two-dimensional, three-dimensional, etc. data as on one-dimensional data. Practically, the spectrum of a two-dimensional signal is first computed over all lines/columns of the signal and than over the other dimension. The result is still a valid spectrum.

Figure 12.3 compares the Fourier transform (first column), the cosine transform (second column) and a Haar wavelet transform (third column, see next section) for one frame of the leading example. Each line displays the image signal reconstructed from a band-limited spectrum at 99%, 90%, etc. of the total coefficients. That is, only the first n per cent of the coefficients were used for reconstruction.

As can be seen, the Fourier spectrum shows an immediate loss of quality. Cosine and wavelet spectra remain almost unaltered. At 50% the Fourier signal appears blurred while the two others are still surprisingly good. At a reduction to 5% cosine and Haar spectra show just the strongest edges. The Fourier spectrum is a mixture of high-level and low-level information. That is, the FT still tries to represent the entire signal while the two other transforms focus on significant intensity changes (edges).

Why is that? Since the application is the same, the difference must lie in the kernels used for Fourier transform and cosine transform.[2] The following equations show the real FT kernel and CT the kernel (simplified).

$$k_{RFT}(x,y) = cos(xy)$$
$$k_{DCT}(x,y) = \cos\left(xy+\frac{y}{2}\right)$$

Adding an additional $\frac{y}{2}$ to the waveforms increases their frequency in the DCT more rapidly (over-linearly) than in the FT. As a result, the DCT collects all low-frequency information (strong edges) in the *first* (lowest) coefficients. Higher coefficients represent just minor changes. The spectrum of a DCT is

[2]The performance of the Haar wavelet will be explained in the next section.

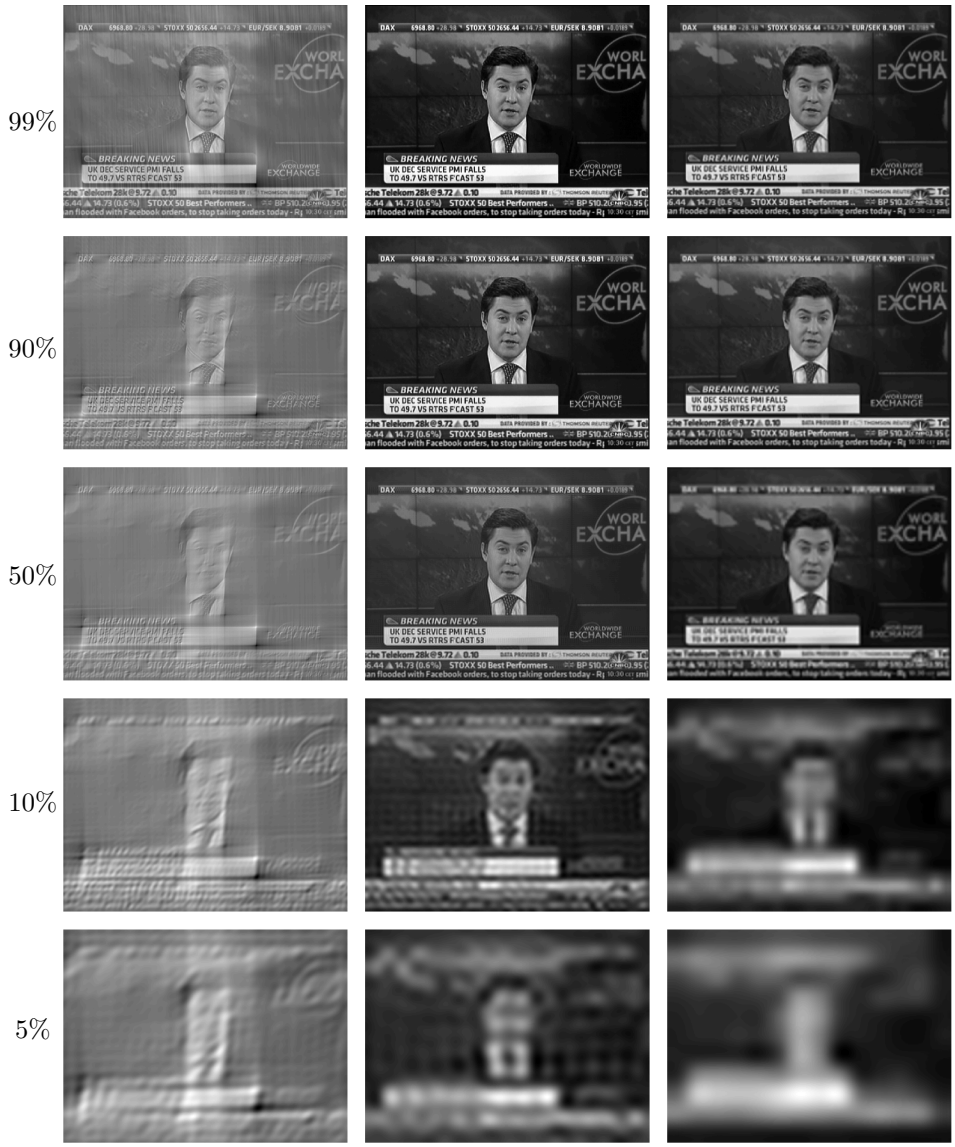


Figure 12.3: Fourier, Cosine and Haar Wavelet Approximation Example for Decreasing Spectrum Size (© *CNBC*).

not linearly organized as a FT spectrum. This transform causes decorrelation between the frequencies by aggregating the fundamental information in just a few coefficients. Therefore, the cosine transformation is tailor-made for *data reduction* by *decorrelation* and, of course, the imitation of visual perception. Before the introduction of wavelets, CT was heavily employed in image and video compression. All these effects are caused by the additional term for the frequency iterator y.

Before we conclude, we have to mention the *Z transform* which is – like the Laplace transform – a superclass of the Fourier transform. Here, the kernel is defined as $k(x, z) = z^{-x}$ and z is typically defined by an expression with one free parameter a. For example, if $z = e^{ia}$ we arrive at the Fourier transform. Particular forms of the Z transform are, for example, employed in biosignal analysis.

In conclusion, Fourier transform and cosine transform are two practically highly relevant discrete transforms that employ continuous base functions. The major difference is that Fourier transform creates a linearly distributed spectrum while cosine transform decorrelates the input signal.

However, the usage of unlimited bases causes problems if time-limited signals (e.g. windows of samples) should be transformed (e.g. artifacts at the ends of the window). Since the 1950ies scientists have developed solutions for this problem. The next section introduces the most important ones.

12.3 Transforms with Limited Bases

The development path to the wavelet transform of today went from the Fourier transform over the *short-time FT* and the *Gabor transform* to the *Haar wavelet*. The latter came into existence before its two predecessors but was first not recognized in its full potential. Below, we discuss the milestones along this path and introduce major wavelet functions.

The *short-time Fourier transform* simply wights the sine base with a windowing function before the application on the input signal. Frequently used windowing functions are the rectangular function, the Gaussian function or its close relative, the Hamming function. Such functions will be discussed in Chapter 14. The kernel weighting operation for windowing function $w(x)$ is performed as:

$$\bar{k}(x, y) = k(x, y).w(x) \tag{12.11}$$

In the 1950ies, Gabor integrated a Gaussian waveform with a Hamming windowing functions, such introducing the *Gabor wavelet transform*. The bottom signal in Figure 12.4 shows a Gabor wavelet. In total, the Gabor kernel has seven parameters, where the simplest parametrization takes the following form:

$$k(x,y) = e^{-\frac{x^2+y^2}{2}}.\cos(2\pi x) \tag{12.12}$$

Applied like a continuous base by positive convolution, the Gabor wavelet was the last step before the development of the *wavelet transform* (WT) with varying base function (*mother wavelet*).

The wavelet transform is distinguished from the standard model of the discrete transform in several points:

- The elements of the base k are time-limited.
- All elements of k are derived from the same mother wavelet and adapted in *scale* and *location*.
- The application by positive convolution is embedded in a recursive algorithm.

The major step in wavelet application is the derivation of a base. The adaptation of the mother function takes the following form:

$$k_{s,l}(x) = 2^{-\frac{s}{2}}\phi(2^{-s}x - l) \tag{12.13}$$

Here, $k_{s,l}(x)$ is the base at scale s and location l. Scales are iterated in multiples of two. The final base consists of linearly independent components. The function $\phi(x)$ is the mother wavelet. The major difference between this type of base and the Fourier base is that the frequency domain does not use the same iterator as the time domain. Instead, scale and location span a two-dimensional spectrum of wavelet coefficients. The coefficients are computed by positive convolution.

$$o_s(s,l) = \langle k_{s,l}(x), o(x)\rangle \tag{12.14}$$

This operation is performed for all relevant scales and locations. The back transformation is almost symmetric to the transformation – of course, it has to iterate over the two-dimensional space. Since the size of the space s, l is not necessarily (and mostly, is not) large enough to cover the entire input signal, *pyramidal coding* is employed to create a spectrum big enough for full back transformation.

Figures 12.4 and 12.5 show the most important mother wavelets. Each mother wavelet is a balanced function, i.e. $\sum \phi(x) = 0$. The first figure summarizes smooth functions that can be used to compute wavelet spectra of smooth signals. The first derivate of the Gaussian function may as well be used to model (soft) edges. The mexican hat function is most sensitive to points (e.g. image pixels, audio peaks). The Meyer wavelet and the Morlet wavelet are very similar

to the Gabor function. The major difference is the frequency of the signal. That is, the Gabor wavelet should perform better on high-frequency information while the Meyer wavelet should be superior for low-frequency information. Of course, the scaling parameter s gives these differences a mostly theoretical relevance.

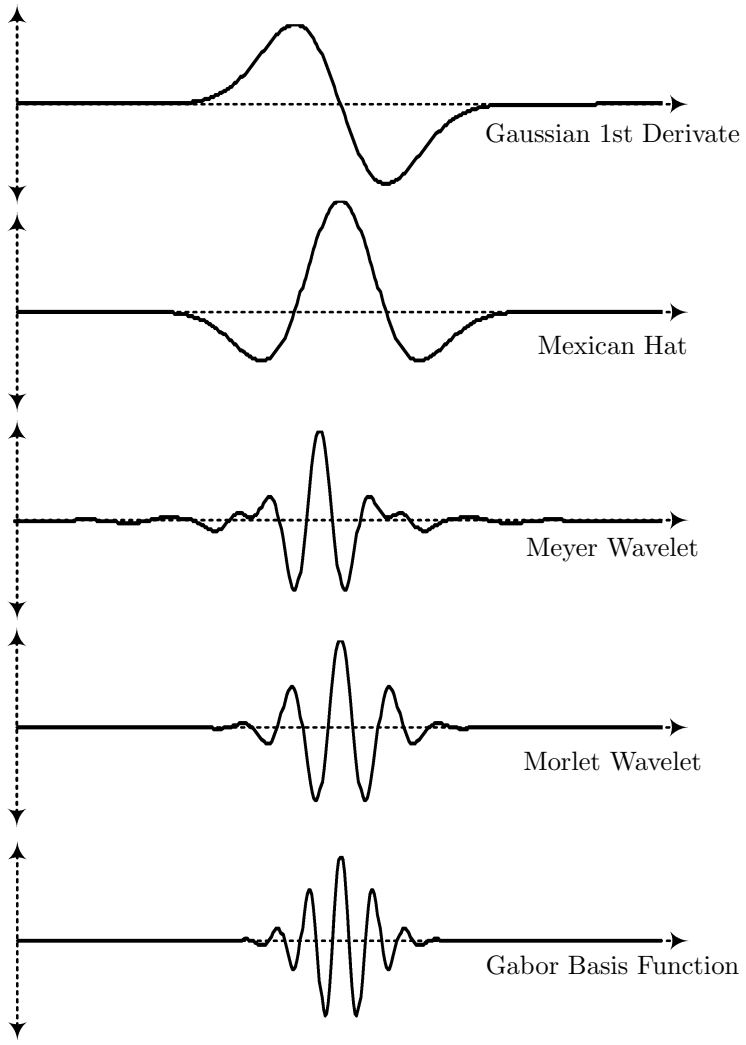


Figure 12.4: Gaussian Mother Wavelets.

The mother wavelets in Figure 12.5 are ideal for edgy signals. The first signal

is the Haar wavelet that was used in Figure 12.3 for wavelet transformation. As we saw, it is very effective for the representation of edges. The second wavelet is an edgy mexican hat that can be employed to represent signals with many isolated peaks (e.g. stock data). This wavelet and the depicted Daubechies wavelet are members of entire families of wavelets. There application lies in spectral transformation of visual and sensor data.

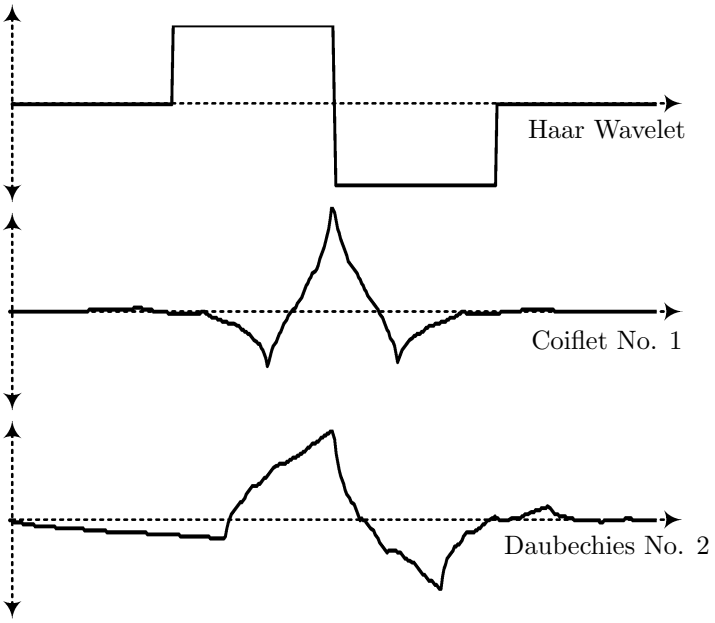


Figure 12.5: Edgy Mother Wavelets.

The actual computation of a wavelet spectrum is fundamentally different than for discrete transforms. Since the base signals are time-limited it is not sufficient to extend the positive convolution over the entire signal and base. Furthermore, the space spanned by scales and locations is normally not as big as the space of the input signal. Therefore, a flexible algorithm is required that creates a spectrum of sufficient size that represents the entire input signal. The solution is a (recursive) *pyramidal algorithm*. Since this algorithm performs the wavelet transform at multiple resolutions, it is also known as *multi-resolution analysis*. The algorithm performs the following steps:

```
o := get_input_signal()
k := compute_base(mother_wavelet)
o_s := 0
```

```
do
   x := high_pass_filter(o)
   o_s := concat(o_s,wt(x,k))

   y := low_pass_filter(o)
   o := downscale(y)
while smooth(o) = TRUE
```

In each iteration, the input signal o is split into low-frequency and high-frequency components, which are fed into the wavelet transform. The resulting spectrum is added to the entire spectrum o_s. The low-frequency components are reduced in size thus creating new high-frequency components. The entire process is repeated until no information is left in the input signal. The execution time of this algorithm depends on the size of the wavelet base k and the complexity of the input signal. Since the function *compute_base* can pre-compute a static matrix of wavelet coefficients, the execution is faster, if the base is larger.

Practically, the algorithm becomes more complex for n-dimensional data. For example, images (two-dimensional) are transformed and scaled down along the lines, columns and both dimensions. That is, on every scale three spectra are computed that are added to the entire spectrum. See the rightmost elements of Figure 13.7 for an example.

The wavelet transform based on mother wavelets and implemented by a pyramidal algorithm has several advantages over total positive convolution.

- The mother wavelet can be chosen as required by the properties of the input signal. The resulting spectrum is optimal for the intended application.
- The base is applied on as many scales and in as many locations as required. This enables localization as well as economic utilization of computing resources (high performance).
- The pyramidal algorithm creates a scale space which is an important component of many media understanding algorithms – in particular for local feature transformations.

The last bullet requires a comment. In recent years wavelet bases have been developed that are very similar to edge operators and point operators (see Chapter 14). The combination of such wavelets with multi-resolution analysis becomes more and more similar to the application of edge operators on scale spaces. We will investigate this issue further in the chapter on local feature transformations.

Before we conclude this section we would like to mention two types of transforms that are with minor differences very similar to wavelet transforms. The first group are *transforms based on polynomials*. The second are *non-separable*

wavelets. Both groups employ bases that are more complex than wavelets derived from a mother function. Hence, these methods may be seen as a link from the domain of wavelet transforms to the domain of template-based matching.

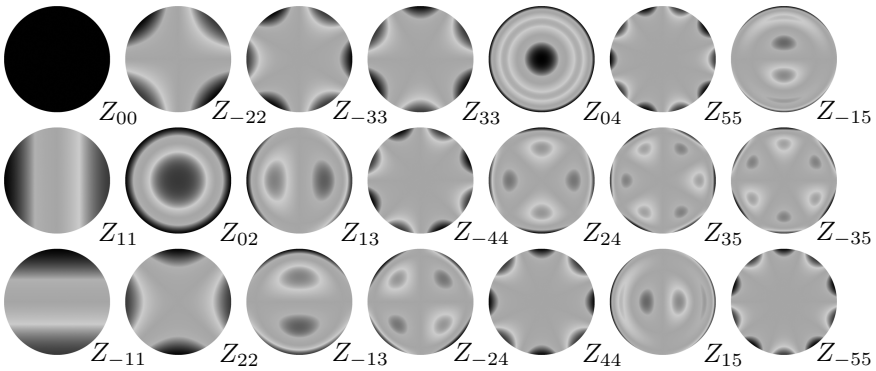


Figure 12.6: Zernike Polynomials Base Functions. This image was created using the algorithm of Claudio Rocchini provided at [309].

Figures 12.6 and 12.7 show two-dimensional polynomial bases. The first are the Zernike polynomials, developed by Zernike for the representation of typical lense defects. Applied by the inner product, the definition of the base can be expressed by the following formula.

$$k_{m,n}(x,y) = \cos(my) \sum_{z=0}^{\frac{n-m}{2}} \frac{(n-z)!(-1)^z}{z!(\frac{n+m}{2}-z)!(\frac{n-m}{2}-z)!} x^{n-2z} \tag{12.15}$$

For $n - m \mod 2 = 1$, $k_{m,n}(x,y) = 0$. Please note that x, y are polar coordinates here. The *Angular Radial Transform* (ART) depicted in Figure 12.7 generates a base similar to the Zernike polynomials. The ART base is defined as follows:

$$k_{m,n}(x,y) = \frac{e^{imy}}{2\pi} \cos(\pi n x)(2 - \delta_n) \tag{12.16}$$

Again, x, y are polar coordinates, δ_n is the Dirac delta function which the peak at $n = 0$. Both polynomials allow the derivation of an infinite number of base elements. In practical application, the first 16 to 64 elements are normally used.

Zernike polynomials and ART are two-dimensional by nature. It is, therefore, not surprising that both transforms are employed in visual media understanding for object representation. The ART, for example, is employed in the MPEG-7

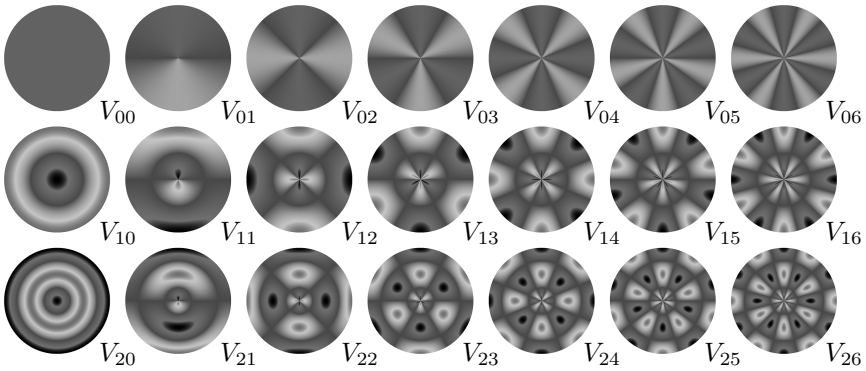


Figure 12.7: Angular Radial Transform Base Functions.

region-based shape descriptor. Both transforms perform well on circular self-similar objects (e.g. faces), because then the similarity between the input data and certain bases is maximal. Hence their application in object description.

Figure 12.8 shows a few examples for one-dimensional polynomials. Others do exist, like the Hahn polynomials or the Meixner polynomials, a generalization of the Kravchuk polynomials. The figure shows only the second and third order elements. As can be seen, the polynomials are continuous but more complex than trigonometric bases. They may be usable for the transformation of data sources with the same properties. One possible application could be the representation of smoothed stock data, another the representation of biosignals. Unfortunately, the potentials of such polynomials has hardly been exploited in media understanding research so far. One exception is the usage of Chebyshev polynomials for image fusion where images are merged by their Chebyshev spectra.

The second class of relatives of wavelets to be mentioned here are non-separable wavelet transforms. Such methods are almost exclusively employed on image data today. In this domain, separable wavelets are able to represent *point features* of the input data well. However, wavelets fail in the optimal representation of media features that are correlated over two dimensions. The reason is the separability property. All dimensions are treated in the same way, i.e. correlation is not considered.

Three examples of non-separable wavelet bases are *ridgelets*, *curvelets* and *contourlets*. Ridgelets, as defined in [46], define a base as follows:

$$k_{m,n,a}(x,y) = \frac{\phi(\frac{x\cos a + y\sin a - n}{m})}{\sqrt{m}} \tag{12.17}$$

Here, a is the angle of rotation, ϕ is some wavelet function and m, n are scaling parameters. The result is a base with properties very similar to the

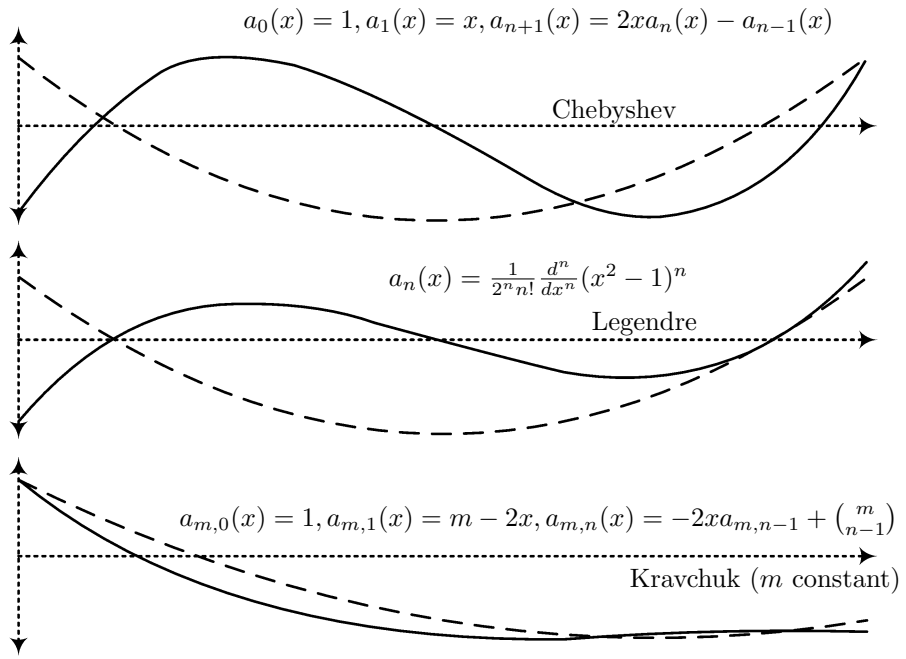


Figure 12.8: Chebychev, Legendre and Kravchuk Polynomials.

two-dimensional polynomials defined above. Similarly, the curvelet approach applies a wavelet on certain sectors of the input data thus emphasizing circular properties in the data. Eventually, the contourlet approach defines an entire scale space for the application of a to be chosen wavelet function.

In summary, wavelets are due to their flexibility (derivation from mother functions, pyramidal coding) state-of-the-art in many media domains. For media understanding, they provide spectra that represent the input data efficiently by a small number of coefficients. As a general rule, *the best wavelet transform is the one with the mother function that is most similar to the input signal.*

Most wavelet transforms guarantee a back transformation. Sometimes, however, something else is desired, for example, rotation-invariant representation of the input data. Then, parametric transforms are the methods of choice.

12.4 Parametric Transforms

We have selected a pair of transforms that are highly important in visual media understanding. The *Radon transform* and the *Hough transform* are both

parametric transforms in the sense that they populate a space and the topology depends on the parametrization of the transform (parameter space).

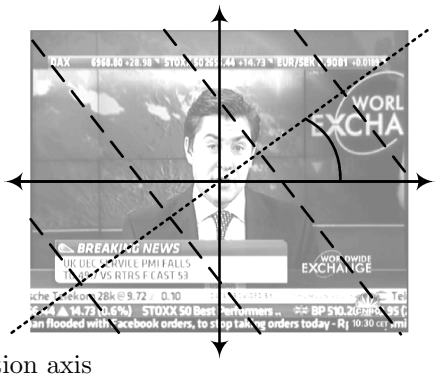


Figure 12.9: The Radon Transformation (© *CNBC*).

Figure 12.10: Radon Transformation Example (black=zero, © *CNBC*).

The principle of the Radon transform is depicted in Figure 12.9. Major element is a projection axis fixed in the image center that rotates and summarizes in every location the intensities of the input media object *perpendicular* to the

location. The result is one vector per angle. Typically, the Radon transform is performed for all discrete angles in the interval $[0, \pi]$.

Figure 12.10 shows the spectra for four different rotations of the same source. As can be seen, the two pairs of 180 degree rotations have the equal spectra. A closer look reveals that the spectra of the pairs are also the same – just shifted by 90 positions. If all spectra were shifted until the globally maximal coefficient is in the first position, all spectra would be equal. That is, the Radon transform creates a rotation-invariant spectrum of the input data.

Figure 12.11: The Hough Transformation (© *CNBC*).

The Hough transformation implements a completely different idea but comes to a result very similar to the one of the Radon transform. Figure 12.11 illustrates the principle. For each sample of the input media object, the gradient (direction and magnitude) is computed. The gradient is the direction of maximal ascent. The computation can, for example be performed by an edge operator or by local neighborhood comparison. See the next chapters for more on this topic. In the second step, an accumulator for direction and offset (the gradient is a vector) is incremented. This process is repeated over all samples.

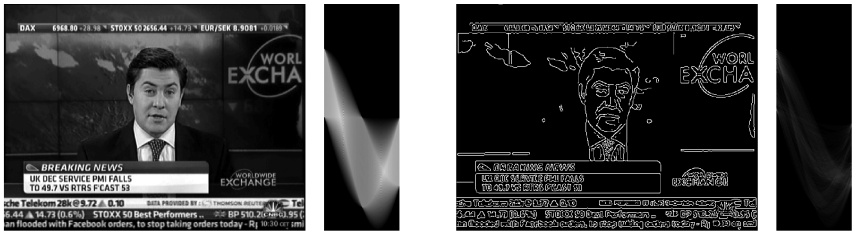


Figure 12.12: Hough Transformation Example (© *CNBC*).

The result, as can be seen in Figure 12.12, is an abstract representation that is – like the Radon spectrum – invariant against rotation. Often, the Hough transform is not applied on the image data directly but on an edge map of the image. the result is the same spectrum, but with fewer elements and smaller peaks.

The Radon transform was developed for the elimination of the influence of rotations. The Hough transform, in the contrary, was developed as an image feature transformation. Rotation invariance is just a side-effect. However, today both parametric transforms are employed for achieving rotation invariance. Some authors even argue that they are equivalent, a point of view we cannot share, because their spectra are obviously different in morphology and magnitude. The common drawback of the two transforms is their bad performance. Therefore, alternative methods have been developed for the achievement of partial rotation invariance. Such methods will be discussed in the subsequent chapters.

In conclusion to this chapter, transforms are an essential element of many feature transformations in media understanding. Spectral representation reduces the negative effect of the gravity of the sample, i.e. the semantic gap becomes smaller. On the other hand, all transforms require significant computation effort, i.e. have a negative effect on the performance. All transforms introduced in this chapter are primarily intended for the interpretation of media data, though some also serve as decorrelation functions (e.g. the cosine transform).

In the next chapter, we see how important discrete transforms are for media understanding. We introduce spectral feature transformations for audio, biosignal, image, stock and video data.

Chapter 13

Spectral Descriptions

Explains the application of discrete transformations for the description of audio and biosignals, discusses the methods employed on visual data and derives a set of methods for the spectral description of stock data.

13.1 Audio Feature Transformations

In contrast to the last chapter, this one introduces practical feature transformations that employ discrete transforms. This chapter continues Chapter 4 from the first part. Like that chapter, this one is organized along the media types. In this section we focus on audio. The next one explains the spectral feature transformations employed on biosignals. Non-surprisingly, those are often similar to the methods employed on audio. The two remaining sections focus on visual material and stock data. The latter domain has hardly seen the application of discrete transforms so far – without good justification, as we think.

The audio section is structured along the four major dimensions of sound perception introduced in the first part. These are *loudness*, *pitch*, *rhythm* and *timbre*. In particular the two latter dimensions can most efficiently be described by spectral transformations. However, before we dive into the pool of features, three pre-requisites should be discussed:

- Spectral aspects of human hearing
- Critical bands and the Barkhausen scale
- Smoothing of spectral windows

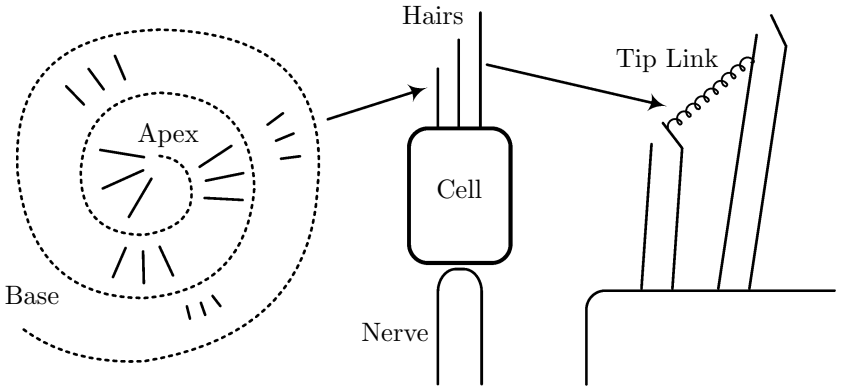


Figure 13.1: Hearing: Cochlea, Hair Cell, Tip Link.

Figure 13.1 shows a conceptual view of the inner ear. Audio waves enter the human ear, are modulated in the auditory canal, propagated by the basilar membrane, malleus and incus. Eventually, the stapes converts the kinetic energy to movement in the fluid that fills the cochlea (left part of the figure). From the base to the apex, the cochlea is punctuated with hair cells (in summary called the organ of Corti). The movement of the fluid causes movement of the hairs on the hair cells. At the base, very short hairs are located which are sensitive to high frequencies. At the apex, the longest hairs can be found, which are best set into motion by fluid movement with low frequencies. At first, it may appear surprising that the longest cells are in the apex of the cochlea spiral. The reason is simply that low frequency movement has higher energy and is, therefore, not prematurely absorbed in the organ.

The hair cell provides the mechanism that converts fluid movement into electrical stimulation of the attached nerve. The right part of Figure 13.1 shows the trick. Every hair cell has a number of hairs which are linked by tip links. The hairs may be seen as tubes of which the covers are opened by hair movement through the tip links. When open, potassium ions can enter the cell from the cochlea fluid. Potassium (K) is a highly reactive alkali metal. When in, the ions cause a depolarization of the cell. In consequence, electric gates at the base of the cell are opened and calcium ions can enter the cell body. The calcium triggers the neurotransmitters at the synapses of the nerve cell which, in return, create a signal in the auditory nerve.

The aspect of this process most remarkable for us is its similarity to the Fourier transform. The hair cells which are ordered in the cochlea by decreasing size and the spiral form of this organ have the same effect as convolution by trigonometric functions has. Specific frequencies are mapped on specific fibres

of the auditory nerve or specific coefficients, respectively. That is, the Fourier transform is a mathematical model of the cochlea organ which is another justification for using this transform in the audio domain.

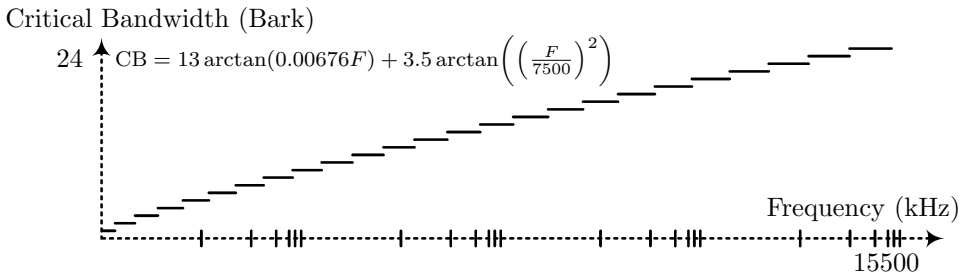


Figure 13.2: Barkhausen Scale with Logarithmic Frequency Scale.

The second prerequisite is related to the first. The *Barkhausen scale* (short: bark) depicted in Figure 13.2 lists the 25 so-called *critical bands of hearing*. The term refers to the groups of hair cells in the cochlea. The length of the hairs does not decrease linearly but in steps from base to apex. Psychophysical experiments have revealed that within groups of (almost) equal size *masking effects* occur. That is, of two simultaneous sound of similar frequency, only the one with the higher energy (loudness) is perceived, the other is dropped (masked). Masking effects hardly occur between groups. Due to this property, each such group of hair cells is called a *critical band*. A critical band is equivalent to a frequency band, i.e. a group of coefficients, of the Fourier transform. Now, the bark scale gives every critical band of human hearing a number from 0 to 24. The equation given in the figure is an approximation. Please note that the frequency scale is logarithmic. That is, the higher the frequency the longer the band and the more similar sounds may be subject to masking. Since masking is an important feature of human hearing, the bark scale is employed in many important spectral audio feature transformations.

The last prerequisite to be mentioned here is a localization issue. As mentioned in the last chapter, the discrete Fourier transform is just an approximation of the continuous definition which goes from negative to positive infinity. Feature transformations, however, operate on small windows of samples. Applying a Fourier transform (or a related transform) on a small window of samples creates artifacts at the borders of that window. Depending on window size and signal content the artifacts may become a significant noise component. The traditional remedy against such artifacts is window smoothing which, practically, means down-weighting the borders of the spectrum. Typically, a Hamming function (similar to the Gaussian function) is laid over the spectrum in order to minimize the artifacts. See the next chapter for more details on window smoothing.

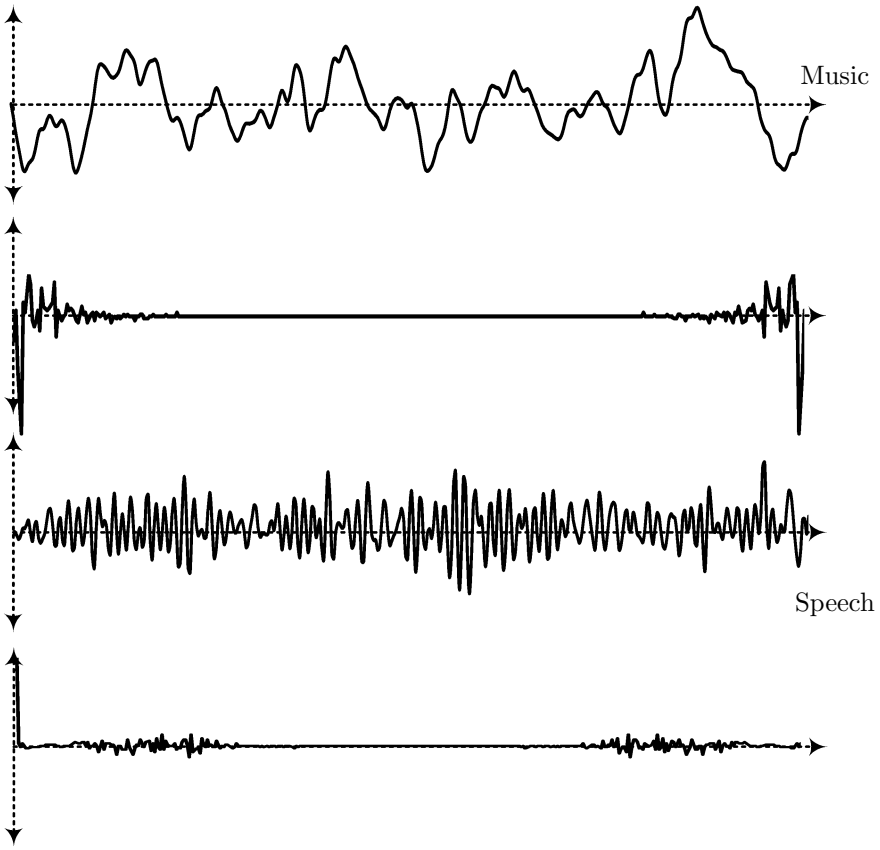


Figure 13.3: Fourier Spectra of Music (top) and Speech (bottom).

For the present chapter we have decided not to smooth the spectra. The natural state of signal and spectrum should, as we hope, support the understanding of the employed methods. Figure 13.3 illustrates the signals and Fourier spectra of the two example audio signals introduced in the first part. As can be seen, the music signal is composed of a few low-frequency bins and a few high-frequency bins. Mid-frequency components hardly occur. The speech signal is similar but a bit more diverse. Knowing that speech is generated in a much simpler way than music, how can that be? The reasons are two-fold. First of all, the depicted music signal shows the same number of samples as the speech signal, but the peak frequency of the spoken word lies around 8000Hz while a music signal can go up 20000Hz. Since the number of samples and the frequency are bound

together by the Nyquist law (see first part), the higher frequencies of the music signal cannot be represented in this toy example. The second reason is the missing window smoothing. The initial artifacts distort the spectra.

Hence, this is the *realistic starting point* for spectral audio description. The question now is: What can we practically do to get a grip on the loudness, pitch, rhythm and timbre if the signal has these characteristics? The remainder of this section gives an answer on this question.

In the first part, we introduced short-time energy as a widely used loudness feature. Indeed are Time-based features are indeed ideally suited for the representation of loudness. The benefit of spectral features in this domain lies mostly in the straightforward band filtering. Arguably the most relevant loudness feature transformation in the spectral domain is the *Sone Feature* (SF). It employs bark scale band filtering and the sone transformation (see Chapter 4 in the first part). The feature transformation consists of the following steps:

1. Localization of the media source by windowing. The window size depends on the type of media and the application. For example, for speech recognition 20ms may be suitable while audio genre classification may require 500ms windows.

2. Fourier transformation of each window. The first and the second step can be performed together in the short-time Fourier transform (STFT).

3. Separation of the critical bands in the spectra. For each critical band:

   (a) Back transformation of the band-limited spectrum into the time domain.

   (b) Application of the sone transformation on the amplitudes.

4. Aggregation of a description from windows and bands per window. If necessary, information filtering by coarse representation or factor analysis.

The algorithm makes clear that SF is an enhanced short-time energy that exploits our knowledge about critical bands and about the perception of loudness. One detail in the computation that sometimes causes confusion is how frequencies (e.g. in the bark scale) are mapped on spectral coefficients. In the case of the real Fourier transform, for example, the spectral coefficients are the result of convolution by the kernel $\cos(yx)$. Since y is the indicator of the coefficient, x is the only free variable which allows for interpreting it as the positional parameter in the cosine function. Then, y determines the frequency of the employed cosine wave and corresponds directly to the frequency requested by the transformation. In short, the indicator of the coefficient is mapped on the frequency directly.

Pitch computation – where the zero crossings rate is employed in the time domain – has the honor of being operationalized by the probably most famous

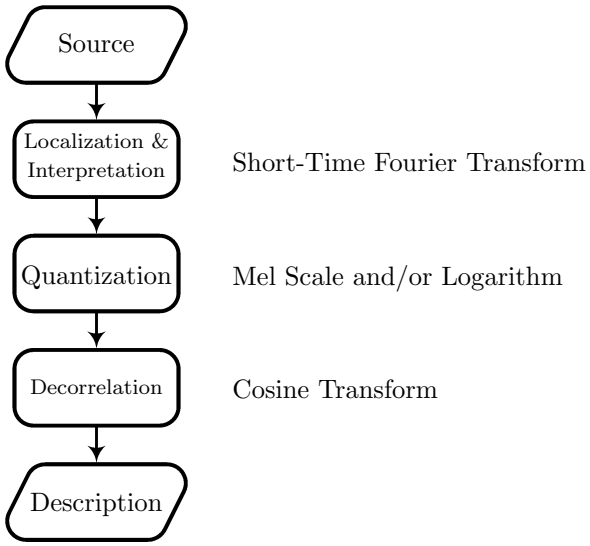


Figure 13.4: Computation of Mel Frequency Cepstral Coefficients.

audio descriptor, the *Mel Frequency Cepstral Coefficients* (MFCC). Since they are so important in audio understanding, their computation is not just explained in the text but also illustrated in Figure 13.4. The MFCC use the majority of important building blocks to summarize an audio signal. In particular, the following steps are taken. First come localization and spectral interpretation by the STFT. On the spectrum, the mel scale is applied (sometimes replaced or augmented by the logarithm). This operation brings the psychology of hearing into the MFCC. In the last step, the cosine transform is employed for decorrelation of the coefficients. The result is called a *cepstrum* (reversed *spec-trum*). The reason may be that in some forms of the MFCC the cosine transform is replaced by the inverse Fourier transform – though this transform has a completely different effect.

Only the first n coefficients of the cepstrum are used as description of an audio window. It has to be noted that MFCC is just a summarizing feature transformation. Interpretation is only performed by spectral representation but not by autocorrelation. Hence, MFCC represent the information in the input media – and that very efficiently due to the cosine transform. However, MFCC do not reflect the signal like a predictive coding transformation would. Still, MFCC are state-of-the-art not just for pitch representation but also for many applications, most noteworthy speech recognition. The practitioner will, whenever working on a new audio understanding problem, take MFCC into the mix

of audio features. Factor analysis shows that MFCC come very close to being an orthogonal base of the input data. This is because of the CT.

In loudness and pitch detection, SF and MFCC are hardly disputed (though many other interesting feature transformations do exist, see [268]). For rhythm detection, however, a number of feature transformations do exist that are closely related. Of these we would like to mention the spectral flux and two forms of perceptual linear prediction. Other interesting approaches such as the beat histogram and the cyclic beat spectrogram are described in [268].

All spectral rhythm feature transformations discussed here start with the STFT and employ some form of autocorrelation (mostly by negative convolution). The *spectral flux* simply performs negative autocorrelation on the spectra of neighboring windows. The windows are typically of fixed size. In consequence, small description elements indicate a *rhythm component* in the respective frequency band. The larger the spectral flux values, the less rhythmic the source. Of course, spectral flux can also be used to measure pitch as the fundamental frequency. Then, the first description elements have to be used, since these represent differences in the lowest frequencies.

Perceptual linear prediction (PLP) implements a plan similar to time-based linear prediction. In detail, the following steps are performed:

1. Perform the short-time Fourier transform for reasonable window sizes.
2. For every critical band, perform psychoacoustic weighting by the mel scale or the logarithm.
3. Perform the cosine transform for decorrelation.
4. Eventually, do autocorrelation by negative convolution.

That is, PLP is highly similar to linear predictive coding (LPC). The major differences are the introduction of psychophysical weighting and of decorrelation by the cosine transform. It appears justified to conclude that here, the success factors of MFCC were applied on LPC. The PLP variant RASTA-PLP adds another logarithmization before the mel scaling which is reversed by applying the exponential function before the cosine transform. The effect is a stronger influence of the mel scaling. Factor analysis shows that the two variants are highly similar. Therefore, we prefer using PLP for rhythm detection.

For loudness, pitch and rhythm detection we have encountered strong feature transformations both in the time domain and in the spectral domain. The same is not true for the description of timbre. Here, spectral descriptions are the state-of-the-art. Three representatives for a large set of descriptions are *brightness*, *sharpness* and *bandwidth* of a timbre. Brightness can be measured by the weighted spectral mean, for example, $\mu(\log(o_s))$. The logarithm decorrelates the

coefficients of an STFT spectrum o_s. The mean will be small if a sound is dark (few low frequencies) or large if it is bright (many high frequencies). Sharpness is often measured as $\mu(w.o_s)$ where w is a weight vector that is sensitive to large differences between neighboring coefficients. Eventually, the bandwidth (richness) of a timbre can be described by $\sigma(o_s)$, the standard deviation of the coefficients. The higher this value, the richer the timbre.

There are dozens of other audio feature transformations that can be employed for audio description. For example, the *chromagram* is similar to MFCC but employs Fourier transform instead of cosine transform, the logarithm instead of mel scaling and summarization instead of averaging. The *MPEG-7* standard defines a number of descriptors for rhythm detection and timbre description. We would like to refer the curious reader to [268] for an excellent survey of many audio feature transformations. The most important ones, though, have been described above.

In particular, MFCC can be employed on almost any problem and where they fail, PLP can be used. In combination with spectral flux, some timbre features and the before-mentioned time-based descriptions each category of audio can be represented by an expressive description vector. In the next section, we will see if the biosignal domain employs similar methods.

13.2 Biosignal Feature Transformations

The content of this section is bound to the time-based feature transformations discussed in the first part and to the preceding section. We investigate opportunities for the application of spectral transformations in the biosignal domain. First, we introduce an important information filtering scheme which is a necessary prerequisite in this domain due to the very noisy input data. Then, we discuss areas of applications for the fundamental transforms, mainly Fourier transform and wavelet transform. Eventually, we go through the list of applications stated in the first part and suggest adequate spectral features.

Advanced information filtering is the topic of Chapter 16. Nevertheless, the biosignal domain requires an exception. Below, we introduce the cross-spectral density as a source separation tool, because – as we explained in the first part – biosignals have an exceptionally bad signal-noise ratio. The simple reason is that one electrode responds to hundreds of thousands simultaneous nervous impulses.

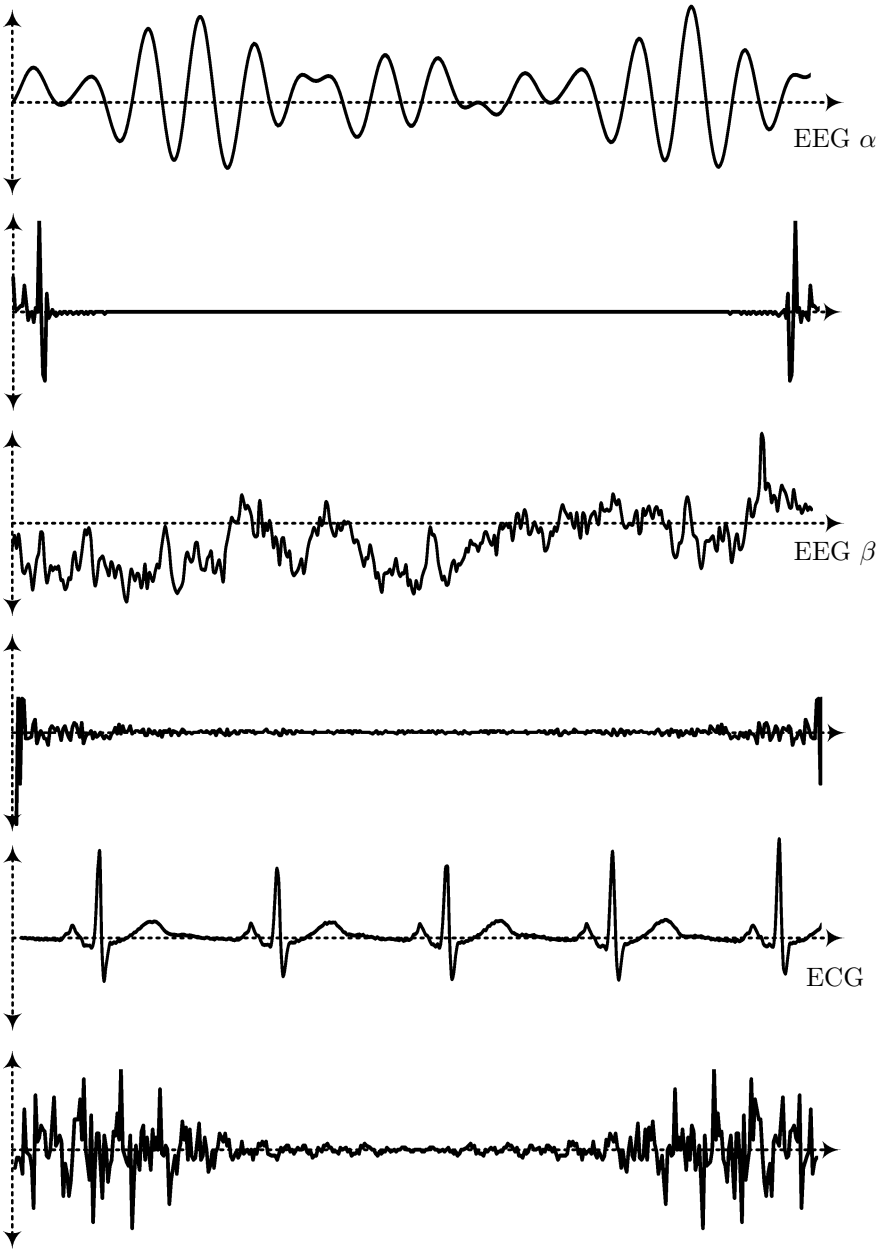


Figure 13.5: Spectra of Biosignals (EEG and ECG).

The *cross-spectral density* (CSD) builds a spectrum from two input signals o_1, o_2. In the biosignal domain, the first source may be the ECG while the second is typically the respiration. Respiration is a noise signal in this setting that overlays the interesting signal, the ECG. The task of CSD is to indicate where (at which frequencies) the noise signal influences the other signal. The CSD χ and the related coherence ρ are defined as follows.

$$\chi_{o_1,o_2}(y) = \sum_{x=0}^{N-1} w_x . r_{o_x,o_y}(x) . k(x,y) \tag{13.1}$$

$$\rho_{o_1,o_2}(y) = \frac{\chi_{o_1,o_2}(y)^2}{\chi_{o_1,o_1}(y) \cdot \chi_{o_2,o_2}(y)} \tag{13.2}$$

Here, r_{o_x,o_y} is the back-transformed covariance of the Fourier spectra of the signals o_x,o_y, i.e. a measure for their *linear dependence*. The w_x are weights from a window smoothing function (e.g. Hamming window) and $k(x,y)$ is the Fourier base. That is, the CSD coefficients are high only where both signals are significant, i.e. where a linear influence of the noise signal exists. Those frequency bands can, for example, be smoothed or the noise can be modeled by an appropriate density function and subtracted from the signal.

Figure 13.5 shows the Fourier spectra for some types of biosignals (without smoothing). As can be seen, the EEG α wave can be described by few coefficients which is not surprising since this wave is defined as being similar to sine waves. In comparison, the EEG β wave generates a significantly more complex spectrum and even the repetitive ECG signal employs more coefficients. Since it is one application of feature transformation of biosignals to distinguish fundamental types of waves, the Fourier spectrum is already a valuable description. For the sake of compactness it can be reduced by statistical aggregation (moments). For example, high variance of the spectral coefficients indicates rather an EEG α wave (subject is resting) than a β wave (subject is active).

Generally, the Fourier spectrum is useful in biosignal understanding where signals are quasi-periodic and smooth. That is, it describes α waves well, while for periodic, but not smooth signals the spectrum becomes unnecessarily complex. For example, the ECG given in the figure can better be described by a wavelet (see below). Simple biosignal descriptions drawn directly from the Fourier spectrum include the *dominant frequency* (maximum of coefficients), *absolute and relative importance* of frequency bands (averaged groups of coefficients) and the size of the *10-90% band* that covers 80% of the input signal. Furthermore, the *periodogram* is a simple transformation of the Fourier spectrum, where every coefficient is taken to the power of two and divided by the number of samples. The periodogram is suitable for distinguishing random signals (uniform distribution of the coefficients) from characteristic signals.

We see that all these feature transformations are significantly less sophisticated than what is applied in the audio domain. However, one transformation of similar design is the *spectrogram*. The spectrogram is computed in the following steps.

1. Short-time Fourier transformation of the input signal

2. Computation of the short-time energy for each window

That is, the spectrogram is high, if the absolute energy in the spectrum is high. The resulting description is similar to the sone feature, though, of course, no psychophysical transformation is employed here. Sometimes, the spectrogram is also computed from wavelets, in particular, Morlet and mexican hat mother wavelets.

As already mentioned, wavelets are used in biosignal understanding for the description of edgy rhythmic patterns (pulses, for example). Such signals are frequently called *spike wave complexes*. The ECG is a typical example. For such waveforms standard mother functions can be employed but it is also tempting to define a pattern tailor-made for the signal of interest (e.g. an ideal ECG pulse). Transforming the input signal by a base derived from such a mother function reduces the analysis of the spectrum to the investigation of non-uniform coefficients. From the global media understanding perspective, this proceeding goes even further into the direction of template matching than ordinary wavelet transformation. The technology is the same, therefore, the labeling of the method is a question of taste.

The last general transform to mention here is the Z transform, which is relatively popular in biosignal analysis. The application is typically in the simplest form, where the resulting coefficients are used as a description directly. Another application is the computation of a spectrum for further analysis by filtering. The Z spectrum is generally interesting for non-smooth non-periodic signals (e.g EEG β waves), but even for these signals some wavelets perform superior.

In the remainder of this section we discuss solutions for specific biosignal understanding problems. Where necessary, we will not just employ the fundamental methods from above but also suggest suitable audio feature transformations. The list of applications is the same as in the first part of this textbook.

One major problem of biosignal understanding is the detection of s*teady-state visual evoked potentials*. That includes the detection of amplitude peaks that correspond to sudden brain activity caused by an unusual visual stimulus. Since the EEG peak occurs usually around 300ms after the presentation of the stimulus the problem is also called *P300 detection*. Traditionally, this problem is approached by wavelet transformation and mother functions that resemble such peaks. Frequently used base functions are the mexican hat wavelet and the

Meyer wavelet. The second is superior for noisy data (more high frequency components). Alternatively, it may be interesting to investigate the cosine spectrum of such a signal which should represent P300 events (that are similar to edges) well.

K complexes which indicate non-REM sleep are distinguished in the EEG by a positive signal peak followed by a negative peak. This type of signal resembles the first derivate of the Gaussian function. Therefore, a wavelet transform with this base is the feature transformation of choice. Interestingly, the literature also suggests using the Meyer wavelet, but we believe that the Gaussian base (if not a Haar wavelet) would create the optimal response in the spectrum. The description could be the first n maxima of the spectrum.

The next domain are *slow cortical potentials*, which express the excitability of brain areas. Technically, they are represented by a low-frequency wave overlay over the EEG wave. In order to identify such waves, a Fourier periodogram with large window sizes can be employed. In a second step – similar to MFCC – the cosine transform can be used for the elimination of all non-fundamental spectral components.

The detection of (the absence of) *changes of oscillatory activity* (COA) is important for the detection of epilepsy. COA are sudden changes in frequency and/or amplitude of a signal. One temporal method for COA description would be the amplitude descriptor introduced in the first part. In the spectral domain, smooth wavelets are well-suited for representation. The frequently used Morlet wavelet could also be substituted by the Gabor wavelet. The description could be constructed from windowed energy values, i.e. a spectrogram. Furthermore, autocorrelation in the form of PLP (without the psychophysical step, of course) could also indicate breaks in the rhythm of the signal.

Eventually, the detection of real or virtual motor activity is a question of the distinction of β waves from γ waves. These two EEG signals have very similar characteristics but a fundamentally different bandwidth. The maximum frequency of β waves lies around 30Hz while γ waves have up to 100Hz. Therefore, the feature transformation of choice is MFCC without mel scaling. This transformation models the fundamental frequency (pitch) which is exactly the criterion of interest here.

In conclusion, we see that the traditional spectral biosignal feature transformations are intended for semi-automatic content understanding. The spectra are visualized and the categorization is left to the expert user. However, most application problems can be automated, for example, by the methods introduced above. In contrast to the time domain, audio features cannot be mapped directly on the biosignal domain. Manipulations motivated by psychoacoustics have to be removed from the recipes. Then way, potent descriptions can be generated that can be made subject to computational categorization.

13.3 Visual Feature Transformations

This section has the same organization as Chapter 5 in the first part. First, we deal with color and texture descriptions and then with shape representation. Feature extraction in the *visual domain* has to take two major differences into account. Signals are two-dimensional and the main source of information is compacted in *edges*, not distributed over the entire signal. Therefore, different operations are employed for spectral representation.

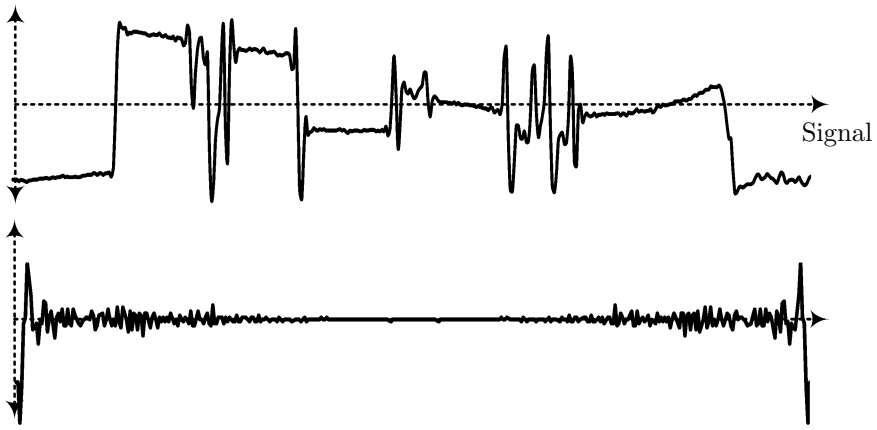


Figure 13.6: Fourier Spectrum of an Image Signal.

Figure 13.6 shows that the Fourier transform does not generate a very informative spectrum for visual information. The signal is the line of the leading example, that goes through the nose tip of the anchorman. The information, down-weighted by artifacts at beginning and end, is distributed over two thirds of the coefficients. The spectrum says, that the signal is a composition of low and high frequencies but the edge information cannot be seen in the data anymore.

What we require is a representation of the contrast in a visual object. In the last chapter we pointed out that the cosine transform provides such a description. Figure 13.7 gives an example. The left column shows the source objects, the middle column their cosine spectrum and the right a wavelet decomposition. If we look at the cosine spectrum first, we see that the spectrum of the original image looks frighteningly close to white noise though on small scale structures are visible. Comparing the original image to the canny edge map, however, shows the value of the cosine spectrum. The spectrum of the edge map is mostly black (zero information) and only the first columns and the first rows contain information. Since the map contains only the edge information of the original

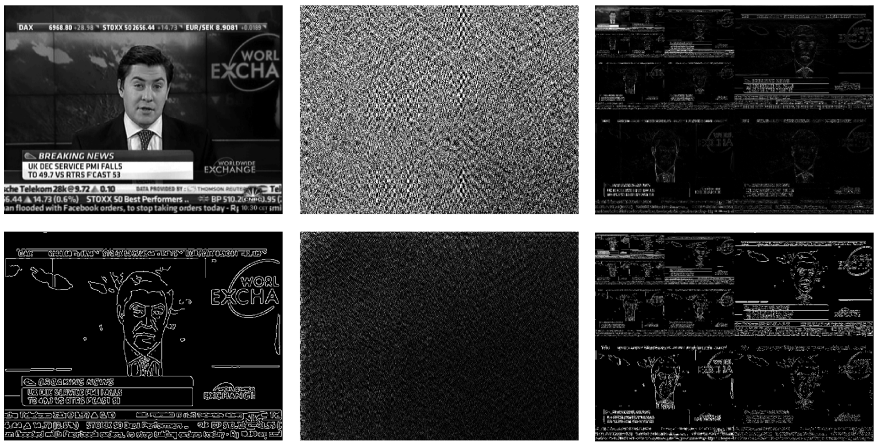


Figure 13.7: Cosine Spectrum (middle) and Wavelet Decomposition (right) of an Image and an Edge Map (© *CNBC*).

visual information we can conclude that these spectral bins represent the edge information. That is, the cosine transform distills the edge information into the first coefficients.

The Haar wavelet decomposition is given as an example for object representation. We can see that – in contrast to the cosine transform – object contours are preserved in the image pyramid that develops towards the upper left corner. As we explained in multi-resolution analysis, the smaller the representation the lower the represented frequencies. The object contours remain visible in the decomposition process because wavelets use limited bases that are transformed to particular scales and, most importantly, positioned at particular locations. Therefore, wavelets appear to provide the ideal spectra for shape representation.

The color domain does not seem to be an area of application for spectral feature transformation that operate on gray-scaled data and, indeed, only few spectral color descriptions have been defined. One exception is the MPEG-7 *Color Layout Descriptor* (CLD) that was already mentioned in the first part of this book. Figure 13.8 shows the extraction process. In the first step, the source object is localized into 64 blocks. The size of the blocks varies depending on the size of the source. In the second step, windows are represented in the YCbCr color model and for each channel (luminance, contrast signals to blue and red) the mean color is computed. Then, for each of the three channels a cosine transform is computed and transformed into a description vector by the illustrated zigzag scan. The result is a description that is half way between color

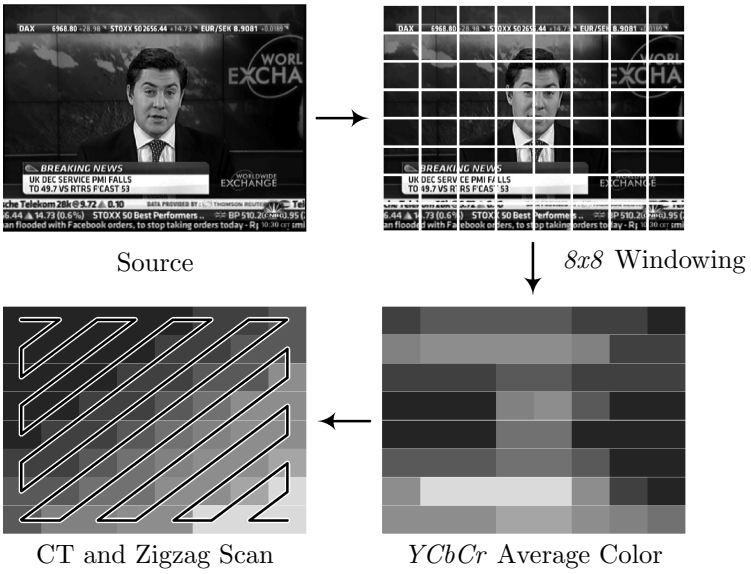


Figure 13.8: MPEG-7 Color Layout Descriptor Computation (© *CNBC*).

and texture descriptions. The majority of information is contained in the first coefficient of the luminance spectrum, which is a texture measure. The remaining coefficients provide only relatively little variance. Still, CLD is an interesting visual description that can be employed – like mel frequency coefficients and zero crossings in the audio domain – to almost any visual understanding problem.

The *MPEG-7 homogeneous texture descriptor* (HTD) employs a two-dimensional Gabor function for spectrum computation. In the second step, the spectrum is localized into $6x5$ windows. Then, for each window, the energy (as in short-time energy) and its standard deviation are computed. The descriptor is an aggregation of the spectral moments supplemented by global energy and deviation. It appears surprising that a smooth function like the Gabor function is employed for spectral representation here. As we saw above, the cosine transform would probably be more efficient. However, HTD does not desire an efficient representation of the visual information but a uniform representation of texture characteristics – something very similar to the timbre in the audio domain. For efficient representation it is preferable to employ the first coefficients of cosine transform instead, i.e. the CLD.

Figure 13.9 gives an example for a completely different approach to spectral representation in the visual domain. The *Euclidean distance transform* (EDT) replaces each sample of a visual object by its Euclidean distance (second-order

Figure 13.9: Euclidean (middle) and City Block (right) Distance Transform (© *CNBC*).

Minkowski distance) to the next edge. The result is a visual spectrum that looks similar to a strong glow effect. In the right part of the figure we see the same transform but computed by the city block distance (first-order Minkowski distance). The major difference is that this approach does not respond strongly to diagonal edges.

The EDT is typically used in media understanding as a pre-processing step of object recognition, because it emphasizes the edge information to an extent that creates compact objects. It may, for example, be used prior to the *MPEG-7 region-based shape descriptor*. This feature transformation applies the angular radial transform (ART) introduced in the last chapter on the input object and uses the spectral coefficients as the description. Of course, this transformation reacts strongly to compact circular objects, for example, faces. Therefore, combining it with some pre-processing object segmentation is reasonable. Alternatively to the ART, the Zernike polynomials could be used which would provide a similar spectrum. Of course, the entire zoo of contour-sensitive methods such as various wavelets, ridgelets, contourlets, etc. could be used as well.

Before we conclude this section, we would like to add a few comments:

- What is the relevance of Radon and Hough transform? These transforms are relevant, wherever rotation invariance is not guaranteed by the method itself. For example, the spectrum of ART or Zernike polynomials is rotation-invariant per se. For some texture features directionality is important to know. Global color descriptions abstract from the orientation anyway. In other cases, such as global object representation by a wavelet pyramid, it is advisable to employ a parametric transform. Furthermore, the coefficients of the Hough transform are considered a texture description in their own right.

- Spectral transformation is also relevant in visual motion description. In Chapter 15 we will see that the properties of the Fourier transform make it an option for the computation of global motion.

- Recently, the *Walsh Hadamard transform* (WHT) gained attention as a spectral transformation of edge information. For spectrum computation, the WHT employs a simple binary base recursively. The approach is equivalent to a Fourier transform applied on an n-dimensional object with binary dimensions. That is, the visual content is interpreted as a high-dimensional block of data. On the other hand, the approach produces results similar to Haar multi-resolution analysis. It is therefore an interesting bridge between the Fourier domain and the wavelet domain.

In conclusion, in the first three sections we have encountered a number of spectral feature transformations that are today employed on audiovisual and other content. Methods of outstanding importance are the Fourier, cosine and wavelet transform. The rest of the recipes is often very similar to the domain of time-based feature transformations. In the last section we try to transfer the idea of spectral feature transformation on a yet untouched domain: stock analysis.

13.4 Spectral Description of Stock Data

Technical chart analysis has hardly seen the application of complex feature transformations, in particular spectral transforms. This is unfortunate, because such methods may be able to extract the non-random part from the fundamental Wiener process. In consequence, spectral transformation could be a valuable pre-processing step in stock description for prediction based on machine learning.

Figure 13.10 shows what happens if we apply the Fourier transform without smoothing on a stock data stream. The artifacts at the borders make the spectrum appear almost uniformly distributed. That would mean that the signal is very close to being random. However, the small deviations can be increased in magnitude by window smoothing and by appropriate feature transformation.

One such method is the periodogram introduced for biosignals. If we take the smoothed spectrum to the second (or, n-th) power we emphasize the specific characteristics covered by the noise of the Wiener process. Knowing further that the low-frequency coefficients are more reliable in a stock signal, since they represent those expressions of the market that are less influenced by daily changes, recommends extracting these coefficients as description input. The actual description could – like a chromagram – be provided by Fourier back-transformation. This feature transformation would predict the fundamental development of a stock like the sliding average or a set of support and resistance lines.

Another biosignal-like approach implements the idea of the recognition of spike wave complexes. Similar to that, primitives like rectangles, triangles,

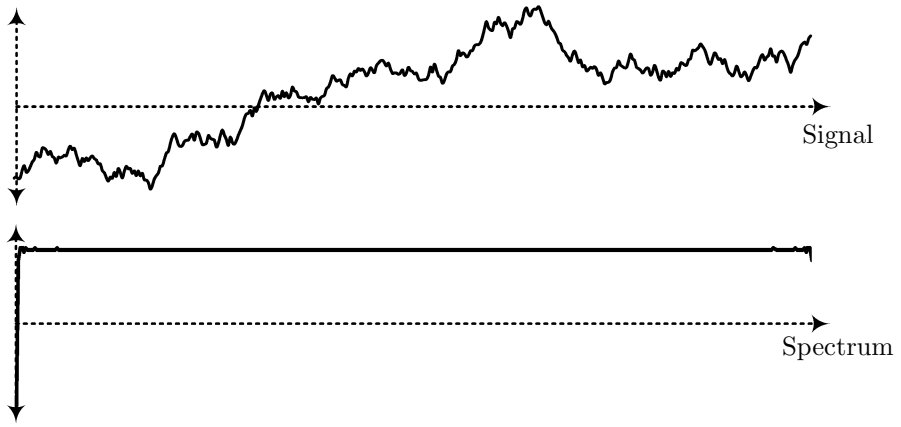


Figure 13.10: Fourier Spectrum of a Wiener Process.

but also more complex forms like butterflies, etc., could be modeled as wavelet mother functions. Then, the coefficients of a wavelet decomposition could be employed as predictive descriptions of a stock.

Generally, wavelets appear to be the ideal solution for short-time prediction in stock analysis. The pattern could be to apply a library of incomplete wavelet templates on the last n samples of a stock by crosscorrelation and interpret the resulting similarity values as belief scores for the likelihood of future continuation of a particular template. It appears reasonable to suggest negative convolution for this type of data because stock values are genuinely integral, i.e. aggregating multiple channels of information.

From the spectral audio toolbox, MFCC, RASTA-PLP, bandwidth and sharpness could be interesting. MFCC – without mel scaling – could be employed to model an incomplete stock (the future is missing). Such descriptions could be compared to prototypes of past and future information by various categorization methods. As for the other domains, MFCC capture the fundamental characteristics of stock data well.

Linear prediction fits naturally to stock data analysis. The added value of RASTA-PLP lies in the increase of magnitude caused by the exponential function. This operation would increase the differences in the spectrum and might lead to better prediction of future developments.

Eventually, bandwidth and sharpness are two timbre features that operate directly on the Fourier spectrum. That is, they cover long-term developments of a signal. With these characteristics, the two feature transformations could be used in the same fashion as MFCC for comparison to prototypes of the future that are motivated by past developments.

In conclusion, many interesting feature transformation could be defined for the stock domain. As always, it is recommended to try everything, apply information filtering and ground truth-based evaluation and choose the methods that perform best. All spectral methods have in common that they reduce the semantic gap by transforming some abstract signal into a set of weights of interpretable base functions. This benefit is paid with relatively bad performance, since spectrum computation requires considerable time and resources. It lies in the nature of spectral features that they are employed on large chunks of data, for example, in order to avoid border artifacts. In the next chapter, we encounter a completely different approach to feature transformation: the description of isolated but characteristic groups of samples. In this domain, spectral features are only of minor importance.

Chapter 14

Description of Local Media Properties

Introduces the scale space approach for windowing, point detection by Hesse matrix criteria, local descriptions by gradients and the transfer of these concepts from the visual to the other media domains.

14.1 General Localization Methods

The local feature transformation methods discussed in this chapter are not distributed uniformly over all media types. In fact, localization to the phrase, word, etc. is trivial for symbolic media. For one-dimensional quantitative signals such as audio, biosignals and stock data the two major forms of localization are windowing and band filtering in the spectral domain. Both approaches have already been introduced. For example, the bark scale can be used together with the Fourier transform for audio band filtering. It is, therefore, not surprising that we focus on the remaining domain: vision. The first three sections of this chapter deal primarily with visual media objects. This section reflects the problem of localization in general. The next section introduces the *interest point* concept for perception-like visual feature extraction. Section 14.3 explains how interest points can be converted into proper descriptions for categorization. Eventually, in Section 14.4 we try to transfer successful methods from the visual domain to the other media types.

This section deals with general solutions for localization. By that we mean approaches for localization that do not contain any form of feature extraction.

Rather, we define loc_i building blocks suitable for combination with other building blocks in feature transformations. The review starts at the well-known rectangular windowing approach and advances over object contour detection to image pyramids and scale spaces, which are state-of-the-art in image and video understanding.

Principally, *localization* in media understanding means to apply the big picture not on a media object as a whole but on certain – interesting – parts (*regions*) of it. Other parts are simultaneously treated in the same way or discarded. The resulting descriptions are merged and made subject to further feature transformation, redundancy filtering or categorization. That is, localization implies cyclic media understanding, i.e. the iterative application of feature extraction on media representations with different levels of precision. The art of localization lies in *differentiating the interesting regions from the rest.*

The simplest solution to this problem is considering everything as interesting and performing rectangular segmentation. We have already come across this form of windowing in the audio domain, where descriptions are generally not computed for an entire media object but for short chunks of time. The localization process is controlled by two parameters, the *window size* and the *hop size* (as defined in the first part of the book). The major advantage of the approach is also its major weakness. The simplicity of the windowing process cannot handle variable borders between objects (one sound, one word, a face, etc.). Semantically related objects are cut in two or more pieces and valuable content is lost. This drawback is of minor importance for audio, because the window size is usually rather small and the descriptions for multiple components of the same object can easily be merged in later steps. In the visual domain, however, the reconstruction is generally significantly harder. Approaches, such as the visual keywords discussed in the first part, suffer from this drawback.

One particular form of occurrence of the border problem are the artifacts created by the transformation of limited chunks by unlimited functions, for example, the Fourier transform. In the preceding chapter, we saw that most spectra contain suspiciously high coefficients at the ends. See, for example, the audio signals in Figure 13.3. These outliers are partially due to insensible cutting of periodic signal components at the wrong point. The general remedy to this problem is *window smoothing.* Figure 14.1 shows three examples. The general approach is very simple.

1. Weight every sample of the media chunk of interest by a window function.

2. Perform the spectral transform on the weighted chunk.

The result is a more or less reduction of the artifacts. The figure shows the results for the speech signal in Figure 13.3. In contrast to the spectrum there, here, we give – for easier comparison – the *absolute values* of all coefficients. As

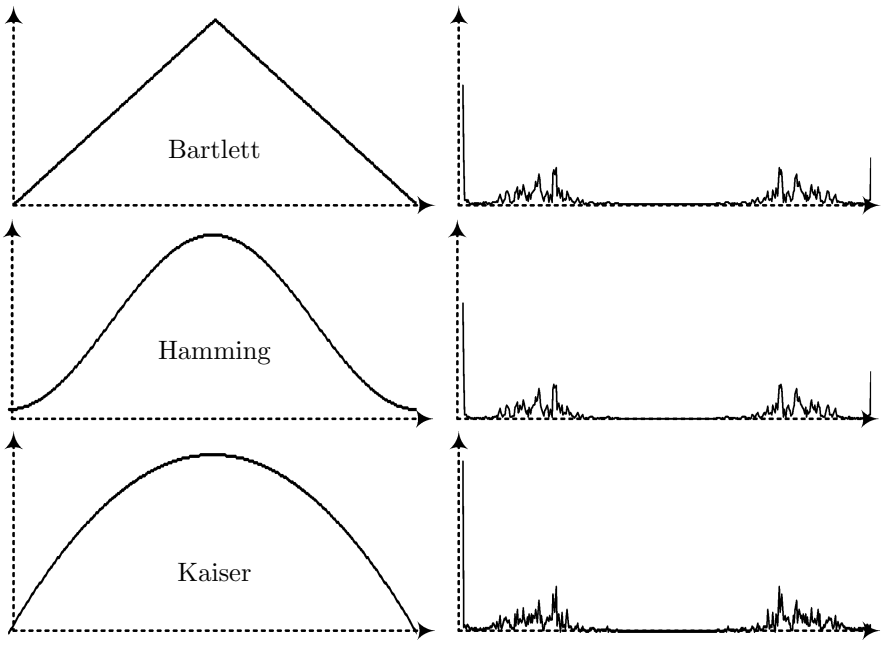


Figure 14.1: Three Window Smoothing Functions.

we can see, all three windowing functions cause a reduction of the initial coefficient. However, the Bartlett and the Hamming function both introduce a new artifact at the end of the spectrum. Still, the Hamming function is considered a good choice for general-purpose window smoothing. For more information on this well-investigated signal processing problem see, for example, [285].

A better form of localization would respect natural boundaries such as visual object boundaries or spoken words. Edge segmentation, as discussed in the first part of this book, endeavors to reach this goal. Edges are assumed wherever large differences of luminance (contrast) are spotted. Alternative approaches include simple thresholding, the split and merge approach and watershed segmentation. In *thresholding* all pixels below a certain luminance threshold are considered non-edges while all others are considered edges. The result is a very crude representation of object boundaries. This method is hardly used anymore. It is only applicable for very simple object detection in controlled, well-illuminated environments (e.g. industrial robotics).

The *split and merge* algorithm combines *region merging* and *region splitting*. These algorithms implement the same idea as agglomerative and separative clustering in cluster analysis. In region merging, neighboring pixels with

similar luminance are merged to objects. The process is repeated until every pixel in a visual object is associated with one object. Region splitting follows the opposite direction. Initially, all pixels are considered one object. From this object, the most unsimilar pixel is removed, and so on. Split and merge combines the bottom-up and the top-down approach for a more balanced result with medium-sized objects.

Eventually, *watershed segmentation* is very similar to thresholding. The idea is that regions with low/high luminance represent valleys/mountains in the relief of the visual object. Hence, from a randomly selected set of starting points neighboring samples with similar luminance are flooded with the intensity average. Repeating this process until all similar samples are made part of an object, results in an object segmentation.

Whatever segmentation technique is used, the result of local description is almost ever better than if the feature transformation was applied on a rectangular grid of windows. In particular, *the interest point methods introduced in the next section produce significantly better results if they are employed on media objects with a contrast-based contour.*

A second problem of localization is the variation of the *resolution* of a visual object. Next to location, the size/clearness of an object matters in the object recognition process of the human sense of vision. We have already encountered the *image pyramid* concept in Chapter 12. An image pyramid can, for example, be computed by multi-resolution analysis (MRA) with some wavelet base. The left element of Figure 14.2 shows a typical image pyramid. Layer 1 represents the largest resolution, e.g. the first iteration of MRA. In layers 2 and 3 the resolution of the object is reduced by one *octave* each. Depending on the context, an octave may be a factor of 2 or some value 2^x.

The major weakness of the image pyramid is the reduced object size in each layer of an object which makes it hard to identify correspondences between objects detected on different levels of the pyramid. The general application of different resolutions is to test the robustness of objects detected on one level by searching them on other levels as well. This application becomes more difficult, if the size of the location set varies over the layers of the media object representation.

The *scale space* approach is similar to the image pyramid concept but without its localization drawback. The right element of Figure 14.2 shows the general idea. The object size is not altered by the process. Instead, the resolution is reduced by applying a blurring filter on the image content. The larger the filter, the stronger the blurring effect. Technically, this operation is nothing else than positive convolution of the visual media object with a template that represents a blurring filter. The most common template is the Gaussian function which is quantized from the following function.

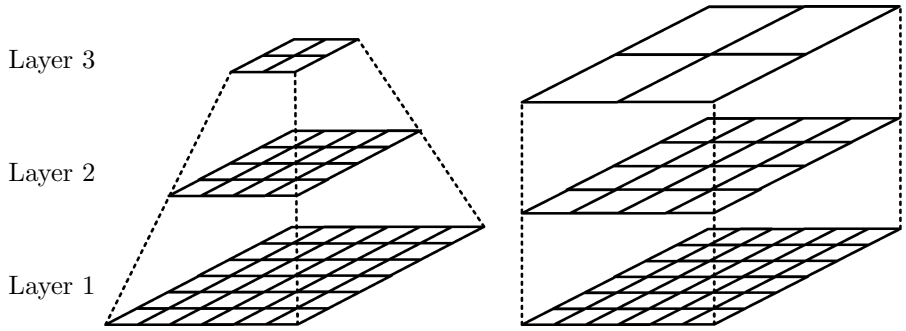


Figure 14.2: Image Pyramid (left) and Scale Space (right).

$$o_{\text{gauss}}(\sigma) = quant\left(\frac{1}{2\pi\sigma}e^{-\frac{l_x^2+l_y^2}{2\sigma}}\right) \tag{14.1}$$

Here, l_x, l_y are the locations in the template o_{gauss} while σ is the scaling parameter that is equivalent to one layer of the scale space. The larger the standard deviation is, the larger the filter becomes. The function *quant* generates a square matrix out of the continuous definition of the Gaussian filter kernel. Of course, the resulting matrix is underdefined, i.e. the continuous kernel can hardly be recovered from the template. The filter for the L_{moore} neighborhood is defined as follows:

$$o_{gauss}(1) = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \tag{14.2}$$

The filter application is simple convolution.

$$o_{ss}(\sigma) = o \otimes o_{\text{gauss}}(\sigma) \tag{14.3}$$

The scale space layer o_{ss} at position σ is created by positive convolution. The object o_{ss} is a complete representation of the scale space.

Scale spaces preserve the locations set of the input object o. On every level, the same points can be addressed. This property is paid with high redundancy in o_{ss}. Neighboring samples are highly similar. The similarity increases with σ. However, this waste of space is outweighted by the advantages of easier application.

Remark: In the last chapter we mentioned the idea that edge operators in combination with scale spaces are very similar to the usage of wavelets (e.g. Haar) and image pyramids in multi-resolution analysis. The discussion so far

should make clear where the similarity lies. The major differences, however, are the divergent location sets and the content of the layers, which are still samples in the scale space while they are wavelet coefficients in the multi-resolution analysis.

Scale spaces are today state-of-the-art for local feature transformation in the visual domain. In the next section, we will see how they are used.

14.2 Visual Interest Point Detection

In this section and the next, we explain how local descriptions are extracted in the visual domain. The present section focusses on the identification of *interesting points* while the next explains the algorithms used to actually *describe* interest points. The most relevant recipes for identification have two ingredients: *scale spaces* – as discussed above – and a specific form of *crosscorrelation* (template matching). This section introduces solutions for the missing ingredients and explains their combination. The theory of the Laplace operator is our leitmotif. Alternative approaches are discussed in place.

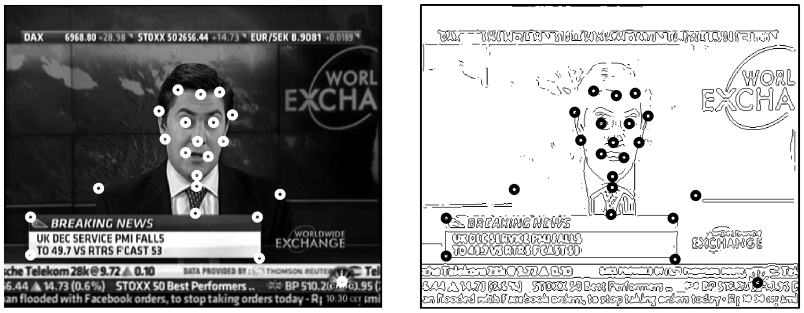


Figure 14.3: Examples of Ideal Interest Points (© *CNBC*).

What is an interest(ing) point in a visual object? This question can hardly be answered on the semantic level of human cognition. The left part of Figure 14.3 makes a few suggestions for the leading example. The nose tip is certainly an important property of a face. So are the eyes, corners of the lips, etc. However, with the limited means of signal processing such points are seldom extractable and describable. Therefore, we use a very simple definition of an interesting point below. An interest point is a sample or a *group of samples* that are *somehow isolated* from their *neighborhood*. Though simple, this definition contains two significant requirements. First, we need some definition of *neighborhood* and grouping. Below, we will mostly rely on the L_{moore} neighborhood of a sample. Second, we need an operationalization for *isolation*. For example, the tip of the

Eiffel Tower is – though connected to the rest of the structure – an isolated point. In this section, we will see, how isolation can be defined effectively.

Before we jump into the details, we should outline the motivation for local description of visual media. The major reason is our desire to imitate human perception. In the first part of the book we saw that the human eye is a scanner that generates a stream of visual information by *saccadic seeing*. Saccades are not uniformly distributed but attracted by certain interesting points (for example, the nose tip). That is, while large parts of a scene are simply ignored, other small elements stand for the majority of the visual information that enters cognition. The author of [12] could show that points with *high curvature* are significantly more important in object recognition than other points. Such points should be recognized as interest points. By local feature extraction we want to achieve exactly this result. Local descriptions should describe the interesting points in detail (e.g. points of change) while ignoring the uniform majority of the visual media information (e.g. edges).

From the technical point of view, we require a better foundation for object representation than edge information. Edges are often noisy, connect areas that are not semantically related and are not invariant against rotation and other transformations – as long as no additional transformation is applied such as Hough transform. Edge detection may be seen as one step in the evolution of local feature detection. Certainly, edges contain valuable information, but the majority of their content is more or less redundant. We are rather interested in the *corners* of edges as one form of interest points. See the right element of Figure 14.3 (canny edge map of the left element) for an example. In the next section we will see that groups of interest points can reach a high degree of invariance against transformations. Therefore, well-defined expressive points are preferable over sketchy edge maps.

For local visual feature extraction we consider a sample isolated, if its gray value (luminance) is significantly different from the gray values of *the majority of its neighbors*. In contrast, an edge could be defined as being significantly different from its neighbors in one major direction but similar in the other. This definition defines an interest point as an extremal point. It is therefore not surprising that the formulation can be operationalized best by computing the first and second derivate of the signal. Figure 14.4 shows an example. The left column of the graph shows the maximum of a continuous signal. Its first derivate has a zero crossing at the position of the extremal point. The second derivate is even more characteristic, because it expresses the maximum in a series of three zero crossings of which the second locates the maximum.

Should we, therefore, define an interest point as a zero crossing in the second derivate of the input signal? No, because in the discrete domain the computation leads to a completely different result. The right column of the figure shows the process. For a sequence of three samples that describe a minimum the first

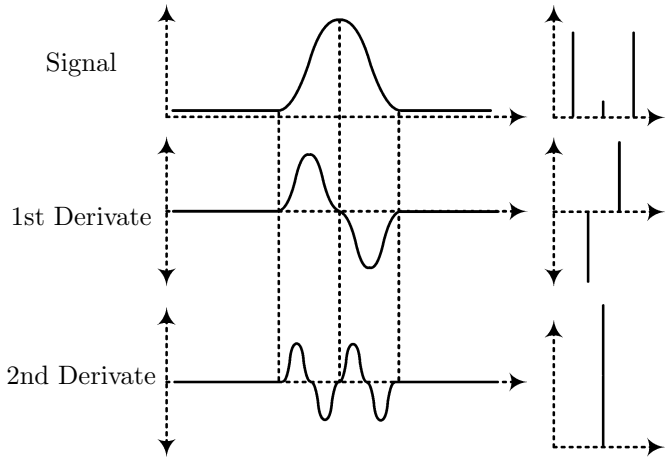


Figure 14.4: First and Second Derivate of a Signal in a Continuous Space (left) and a Discrete Space (right).

derivate has a zero crossing in the – non-existent – center point, which leads to a *maximum in the second derivate*. This is because the partial derivation in the continuous space is translated to taking the contrast in the discrete space. The following equation shows the computation for some media object o.

$$\delta_x = o_{l_x} - o_{l_{x+1}} \tag{14.4}$$

Here, δ_x is the derivation in direction x at position l. For the sake of simplicity we omit the location parameter in δ. Still, for the understanding of the rest of the section, it is important to keep in mind that δ always refers to a concrete sample at location l. Using the above equation, arbitrary partial derivates can be computed by the following rule.

$$\delta_{xy} = \frac{\partial o}{\partial x \partial y}\Big|_l \tag{14.5}$$

That is, the second derivate is computed by twice applying the derivation in direction x. The individual derivations are separable.

The following example should make the practical application of the derivation operator clear.

$$o = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

Media object o is an image of three rows and three columns. Each variable $a-i$ stands for a luminance value. For this object, the first and second derivates in directions x, y can be computed at center position as follows.

$$\delta_x = \begin{pmatrix} d-e & e-f \end{pmatrix} \qquad \delta_{xx} = d+f-2e$$
$$\delta_y = \begin{pmatrix} b-e \\ e-h \end{pmatrix} \qquad \delta_{yy} = b+h-2e$$

We see that the second derivates are both scalar values. This is the practical reason why interest point detection is usually performed in 3x3 Moore neighborhoods.

Now, it is clear that we want to express isolated points by their δ_* properties. The straightforward solution would be to look for points that are maximal/minimal in both directions x, y. Generally, *interest point criteria* can be derived from the Hesse matrix.

$$o_{\text{hesse}} = \begin{pmatrix} \delta_{xx} & \delta_{xy} \\ \delta_{xy} & \delta_{yy} \end{pmatrix} \tag{14.6}$$

Similar to the Jacobi matrix that assembles all first derivates for a function, the Hesse matrix assembles all second derivates. We use it as the ground for the definition of three popular interest point criteria b.

$$b_{lap} = \delta_{xx} + \delta_{yy} \tag{14.7}$$
$$b_{doh} = \delta_{xx}\delta_{yy} - \delta_{xy}^2 \tag{14.8}$$
$$b_{har} = \delta_{xx}\delta_{yy} - \delta_{xy}^2 - \delta_{xx}^2\delta_{yy}^2 \tag{14.9}$$

The first form b_{lap} is the *Laplacian* approach (also known as the Nabla operator ∇). It uses simply the trace of the Hesse matrix. The second approach b_{doh} is called the *Determinant of the Hessian* since it computes the determinant of the Hesse matrix. The last one, b_{har} is an advanced form of the *Harris corner detector*.[1] Before we discuss the similarities and differences of these three criteria, we have to name their common optimization condition.

$$|b_*| \rightarrow \max \tag{14.10}$$

That is, we believe a sample to be an interest point, if the absolute value of its criterion is a maximum. The absolute value is required for the Laplacian, the two other criteria generally produce maxima.

[1] Often, the second and third terms of the Harris detector are downweighted by a factor w. For the sake of simplicity this degree of freedom is omitted in this general introduction.

Remark: The expert reader may be surprised that we do not distinguish between *corner detection* and *blob detection*. The first problem is usually approached by methods based on the *autocorrelation matrix*, i.e. the first derivates around a given point while the Hesse matrix is usually employed for blob detection. For two reasons we do not follow this line. Firstly, it is a fact that all operators introduced here can as well be employed for corner detection as for blob detection. The difference lies only in the embedding of the operators in scale space which can be performed for all operators equally well. Secondly, in the highly constrained case of Moore neighborhoods of digital samples the actual difference between the autocorrelation matrix and the Hesse matrix exists only in the δ_{xy} element. Practically, the autocorrelation value of this component has similar statistical properties. The small difference in absolute values can easily be compensated by a weight for subtracted diagonal components (as in the case of the Harris detector).

For the understanding of the Harris corner detector it is important to know that its evolution has led to two forms. The second form looks for a maximum in the Eigenvalues of the Hesse matrix.

$$\lambda_{1,2} = \frac{\delta_{xx} + \delta_{yy}}{2} \pm \sqrt{\delta_{xy}^2 - \left(\frac{\delta_{xx} - \delta_{yy}}{2}\right)^2} \tag{14.11}$$

This result for the Eigenvalues λ_* is reached by transforming the Eigenvalue problem of the Hesse matrix to a square function and solving this function. The Eigenvalues are maximal if the second derivates in both directions are high (first term), if there is no diagonal relationship (second term δ_{xy}^2) and if the two partial derivates are very similar (last term). This reasonable definition is put to the extreme by the b_{har} definition above, which is maximal if the derivates are high in both directions, no diagonal components exist and the deviations between the derivates in the main directions are as small as possible. Since the latter definition uses multiplication where the first uses summarization and contrast, it reacts much stronger to small deviations. It is, therefore, stricter.

In contrast to the Harris detector, the Laplacian detector does not consider diagonal edges and does not give a penalty for unbalanced differences in the both major directions. The Laplacian is the least strict interest point detector in our list. The determinant of the Hessian, eventually, considers diagonal edges negatively but does not give a penalty for unbalanced derivates in the main directions. This criterion lies in the middle between Laplacian and Harris detector.

The practical selection of an interest point criterion depends on the desired strictness, but as well on the available resources for computation. Of the three presented criteria, the Laplacian detector can easily be operationalized by template matching. If we define the Laplacian criterion for the example above, we

get the following result.

$$\delta_{xx} + \delta_{yy} = b + d + f + h - 4e \tag{14.12}$$

The same result can be achieved by applying the first of the following two matrices on the media object o by positive convolution: $b_{lap} = o \otimes o_{lap}$.

$$o_{lap} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{14.13}$$

The first matrix is equivalent to the formal derivation of the Laplacian criterion. However, practically the second form is the more frequently used one. It considers the diagonal points of the neighborhood and is therefore similar to the determinant of the Hessian approach.

Apart from the named criteria for interest points a few others do exist that are not based on the Hesse matrix but have also proven to isolate interesting points. The *FAST approach* investigates a circular neighborhood (the so-called Bresenham circle, i.e. a digital approximation of a circle, only points over the circular contour are considered) around location l and assumes this sample to be an interest point if its intensity is $n\%$ higher/lower than the neighboring samples. Of course, the quality of this approach depends on the tuning parameter n. The *SUSAN approach* extends FAST by down-weighting neighboring pixels by a Gaussian function.

Figure 14.5: Gradient Vector Components in Ridge Detection.

Eventually, one recent development in the field of interest point detection is *ridge detection*. The ridge is a concept somewhere between interest points and edges. Figure 14.5 shows an example. Ridges are longitudinal objects distinguished by their luminance. Like interest points, ridges can nicely be defined using the Hesse matrix. The processing is to rotate the Hesse matrix until the diagonal components δ_{xy} are zero. This result can, for example, be reached by replacing the matrix with its Eigenvectors. Then, ridges are defined as follows.

$$\delta_{\bar{x}} = 0 \wedge \delta_{\bar{x}\bar{x}} < 0 \wedge |\delta_{\bar{x}\bar{x}}| > |\delta_{\bar{y}\bar{y}}| \vee \delta_{\bar{y}} = 0 \wedge \delta_{\bar{y}\bar{y}} < 0 \wedge |\delta_{\bar{y}\bar{y}}| > |\delta_{\bar{x}\bar{x}}| \tag{14.14}$$

Here, $\bar{x}, \bar{y}$ are the rotated directions of the Hesse matrix. Of the two or-connected expressions the first and second terms define ordinary maxima. The third terms express the longitudinal aspect. In the direction of the maximum the elongation must exceed the one of the other dimension. Ridges are certainly local features. However, their current major application lies in the description of high-level semantics such as symmetries. We will, therefore, meet them again in the third part of the book in Chapter 25.

So far, the section can be summarized as follows. Interest points are practically detected by computing the Hesse matrix for each non-border sample, computing a criterion for each point and selecting those points as candidates that are extreme. Apart from the problems of how to evaluate the quality of candidates and how to describe interest points (both of these questions are answered in the next section) one obvious question remains. The current state of affairs allows only the detection of one-sample interest points. *How can we detect larger blobs as interesting* (for example, the ball in a football game)?

The solution to this problem is a straightforward application of scale spaces. Most state-of-the-art interest point detection methods employ the following algorithm.

1. Compute a scale space o_{ss} for the input media object o, for example, using a Gaussian function.

2. Employ the interest point detection algorithm on each level of the scale space.

3. Select those samples as candidates that fulfill the optimization criterion on at least n layers.

Since scale spaces preserve the locations set on all layers, corresponding interest point candidates can easily be matched. This algorithm represents the general scheme of local feature extraction in the visual domain. Apart from its simplicity it has in advantage that the scaling step and the detection step can be merged if for both convolution operations the templates are given as matrices. For example, if smoothing is performed by o_{gauss} and detection by o_{lap} than both operations can be merged to $o_{log} = o_{gauss}.o_{lap}$ where the dot denotes point-wise multiplication. Where necessary, the size of the Laplacian template has to be increased. This approach is called the *Laplacian of Gaussian* method.

One noteworthy exception from this interest point detection scheme is the *Maximal Stable Extremal Regions* approach (MSER). MSER neither employs a standard scale space nor detection by convolution. Instead, the following algorithm is applied on an input object o where each sample is a luminance value in the interval $[0, 100]$.

```
for t:=0:100 do
   h(t) := o;
   foreach l in L(o) do
      if (h(t,l)<t) then
         h(t,l) := 0
      else
         h(t,l) := 1
      endif
   endfor
endfor

m:=0
for t:=1:99 do
   m(t) := sum(h(t+1),h(t-1)) / sum(h(t))
endfor
```

Here, L is a set of locations, t is a threshold and h is a space of binary images for increasing luminance thresholds. Function *sum* simply counts the sum of all values in an image. Eventually, m holds a value for each non-border layer that is minimal if the Gestalt of a region (samples with value '1') does not change over three consecutive layers. Such a layer – identified by index t – is called an MSER. Of course, the algorithm can easily be extended to larger spans of stability by considering more layers in the calculation of m.

The MSER algorithm is very simple yet highly effective. It covers the scale space effect by identifying blobs through varying thresholds of luminance and it covers the detection-by-isolation idea by comparing neighboring layers. The result is a reliable interest point detector.

In conclusion to this section, the main path towards visual interest point detection today is computing a Gaussian scale space and applying an optimization criterion on each pixel that employs the second derivates in form of the Hesse matrix. In the next section we will see, how such interest point candidates can be tested for validity and robustly be described by their neighborhood.

14.3 Local Descriptions of Visual Media

The description of visual media objects by local properties comprises two steps. In the first, the candidates for interest points have to be filtered and described in an expressive form. In the second step, a model has to be defined for the comparison of sets of interest points in the categorization process. The second step is required for fitting the interest point concept into the big picture of media understanding. Below, we deal with the filtering and description problem first and, in the second part of the section, with the comparison problem.

Why can we not just use the location of an interest point as its description? One technical reason is that locations are simply too short for providing a proper data structure for the storage of semantic variance. More relevant, though, is the question: What characterizes an interest point? It is certainly not its location. For example, the position of the top of the Eiffel Tower will vary significantly in the photos taken by tourists. Still, it is clearly recognizable. What characterizes an interest point is its neighborhood – hardly surprising, since we located them by local media properties in the last section. Therefore, *interest points should be described by the distinct properties of their neighborhood.*

In recent years, a number of recipes have been suggested for the description of interest points. Most of them employ *intensity gradients* – though practically highly relevant exceptions do exist. One classic approach from this group is the *Scale-Invariant Feature Transform* (SIFT) algorithm. In the next paragraphs, we describe this approach. Then, we illustrate alternative approaches by emphasizing their differences to SIFT.

The SIFT algorithm as it was originally defined in [237] and refined in recent years employs the steps illustrated in Figure 14.6. As we can see, the recipe consists of one localization building block – the point detection as described in the last section –, a straightforward interpretation step for the creation of the neighborhood-based description, and several intelligently, yet heuristically defined quantization steps.

The detection of interest points is performed using the *Difference of Gaussians* approach which should approximate the Laplacian of Gaussian approach. This procedure does not apply template matching with the Laplace operator and positive convolution. Instead, negative convolution is used to compare a spatial location on one scale to the same points on neighboring scales. Differences beyond a certain threshold are considered interest point candidates.

This procedure leaves the algorithm with a large set of relatively unstable interest point candidates. In the interpolation step, the location of each candidate is refined by computing the first and second derivates in directions x, y, σ. Here, σ is the resolution, i.e. the location in the scale space dimension. Practically, a second-order Taylor expansion is applied and the new location is defined by the following equation.

$$\bar{l}_{x,y,\sigma} = l_{x,y,\sigma} + \delta_{x,y,\sigma} + \delta\delta_{x,y,\sigma} \tag{14.15}$$

The new location $\bar{l}$ is moved in the direction of the first and second derivates $\delta, \delta\delta$ – two delta coefficients. If the change in location is above a certain threshold, the new location is considered a better starting point than l and the refinement step is repeated for the new location $\bar{l}$.

After interpolation, weak candidates are removed. A candidate is weak if it has low contrast to its neighbors or if it is part of an edge (ridge). Low contrast

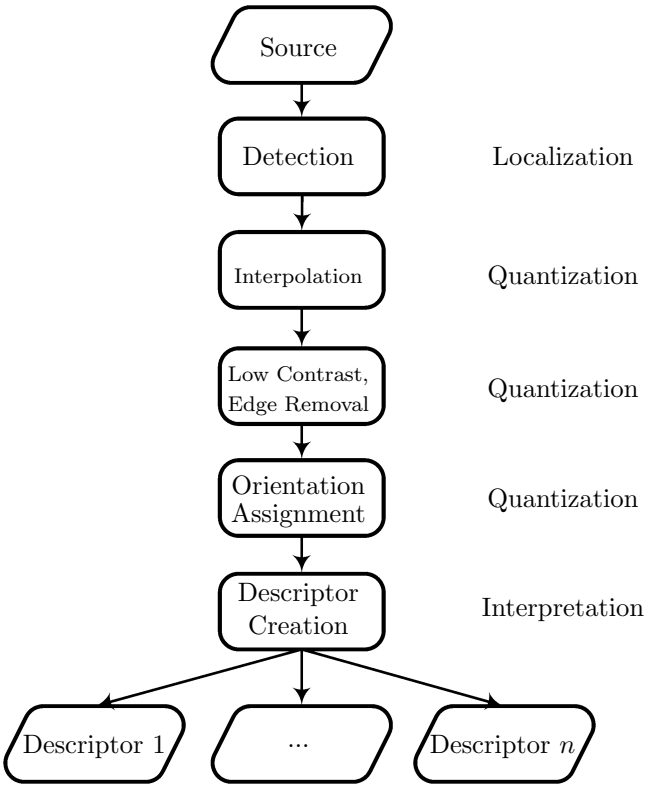


Figure 14.6: Scale Invariant Feature Transform.

is detected by computing the second-order Taylor expansion in directions x, y (the scale remains constant). If this simple gradient is below a certain threshold, the candidate is discarded. If not, its spatial location is adapted by the gradient vector. Edges are recognized by comparing the Eigenvalues of the Hesse matrix at location l as in ridge detection. However, the application goes into the opposite direction. If the relationship of the two Eigenvalues is *above* a certain value – which indicates a ridge/edge – the candidate is discarded otherwise made subject to the orientation assignment step.

As we can see, the quantization so far is based on a number of heuristically defined thresholds, which is both good and bad. It is good since due to the thresholds, SIFT interest points can be computed quickly and with high reliability. The bad aspect is the rigidity of the thresholds which are not adaptable to the requirements of different applications. Figure 14.7 shows an example. For the media source in the top left a high number of candidates is computed of

Figure 14.7: SIFT Example (© *CNBC*). This figure was created using the software library *vlfeat* in version 0.9.9 [382].

which only the few interest points in the upper right element remain. As we can see, the face of the anchor person is represented by just one reliable interest point – the nose tip. More points would be desirable in this region. This feature is currently not part of SIFT but could, for example, be realized by defining different profiles for different media regions (e.g. detected faces).

In the orientation assignment step, the main directions of the neighborhood of the interest point are detected by gradient computation. In detail, the following algorithm is performed for all neighboring points of location $\bar{l}$ in the media object o, i.e. $\theta(o, \bar{l}, \epsilon)$.

1. Compute direction a_1 and magnitude a_2 of the steepest ascent (*gradient*) of the point as follows:

$$a_1 = \sqrt{\delta_x^2 + \delta_y^2} \tag{14.16}$$

$$a_2 = \arctan2(\delta_x, \delta_y) \tag{14.17}$$

Here, the δ values denote the first derivates in the given directions.

2. Quantize the orientation to 36 bins by $\bar{a_2} = round(a_2/10)$.

3. In the histogram of directions add the value $a_1 * 1.5\sigma$ to the bin of $\bar{a_2}$. The value σ is the identifier of the scale space layer.

The result is an orientation histogram in which the highest peak identifies the main direction. For this direction and all peaks that are within 80 per cent of the highest peak, a descriptor is computed. This algorithm creates a certain amount of rotation invariance.

In the last step, one description is computed for each selected direction by the following algorithm.

1. At the location and scale of the interest point a 4x4 grid of 4x4 samples (=16 regions) is laid over the scale space. This grid is rotated into the direction of the peak.

2. For each region, an orientation histogram as in the orientation assignment step is computed. The only difference is that only eight directions are distinguished. The bottom row of Figure 14.7 illustrates such orientation histograms.

3. The resulting 128 bins (8 bins per histogram, 16 regions) are smoothed by a Gaussian, normalized to unit length, denoised by removing all bins that are close to zero and normalized to unit length again.

Eventually, the SIFT algorithm describes each interest point in one direction by 128 values. The entire media object is described by a relatively small number of such *keypoints* that are robust against changes in illumination, rotation and other transformations. This recipe has been so successful that it is today also used as a global feature transformation. The global SIFT (also known as GIST) employs the description algorithm not on the neighborhood of one interest point but on an entire media object.

SIFT is only one – prominent – example for local feature transformation. Alternatives include ColorSIFT, GLOH and SURF. The *ColorSIFT* algorithm takes colors into account and extracts interest point candidates only at color edges which is reasonable, because we usually distinguish objects by their colors, not their luminance. The *GLOH* algorithm is highly similar to SIFT but employs more scales for detection, and factor analysis for the reduction of description size. *SURF* employs the insight that scale spaces in combination with local operators are very similar to wavelet decomposition. The algorithm uses a Haar wavelet decomposition instead of the scale space approach in SIFT. Of course, the gradient-based feature transformation introduced above is only one way to

describe local media properties. Recent experiments have shown that applying state-of-the-art color and texture features on local neighborhoods often results in descriptions that are as good as or even better than SIFT descriptions. That is, the interest point detection step should be seen decoupled from the description step. See [325] for a nice performance comparison of local description methods.

Furthermore, it could be interesting to involve higher-level semantics in the quantization sequence of the feature transformation. One option would be to employ the laws of Gestalt for the elimination (or protection) of interest point candidates. For example, if some interest point candidates indicate a Gestalt primitive such as a group of aligned elements, then fitting points with too low contrast should still be considered while non-fitting points with high contrast should be eliminated. Implementing this idea would require the definition of trade-offs between different interest point criteria that would turn the simple SIFT recipe into an intelligent yet to be parameterized algorithm. Despite the curse of dimensionality, introducing models and templates for some important applications might be worth the effort.

One problem remains: Though the size of each descriptor is fixed, the total length of a description depends on the number of selected interest points. Even worse, the sequence of interest points depends on the viewpoint. That is, it is not possible to define a fixed-length description of local properties where each description element has a clearly defined meaning. Therefore, the big picture of media understanding is not directly applicable. We require an intermediate step that transforms the cloud of descriptions into a well structured media representation.[2]

This intermediate step is a generalization of the visual keywords approach introduced in the first part of the book. There, we extracted randomly chosen regions of fixed size and compared two sets of regions by dynamic association. Choosing region centers randomly implies the risk that important objects are cut at undesired points. This risk is minimized by the careful selection of interest points as peaks in some neighborhood.

Remark: From a formal point of view, we can argue that one important step is missing in the SIFT recipe. All feature transformations we discussed so far finished with an aggregation step as the complementing element to initial localization. SIFT lacks this step. It has to be added prior to categorization.

The generalization of the visual keywords approach typically used on local descriptor sets is called the *Bags of Features* (BOF) approach. It was derived from the *Bags of Words* approach in text understanding. Figure 14.8 illustrates the idea. For two sets of descriptions a three-step algorithm is executed. In the first step, descriptions of interest points (words) are quantized to members of

[2]This topic is half way between feature extraction and categorization. Since it only appears for local visual features, we decided to discuss it here and not together with the general-purpose methods discussed in the categorization chapters.

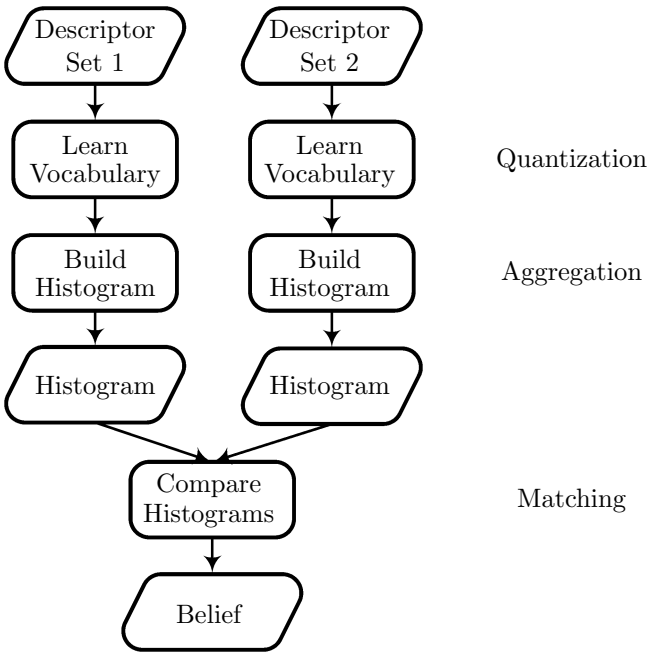


Figure 14.8: Bags of Features Method.

a so-called vocabulary (for text, for example, the principal forms). The members of the vocabulary are used as bins in a histogram. The histograms are populated by counting the occurrences of all members of the vocabulary. Eventually, the histograms are compared by some similarity measure. The last step can, of course, be replaced by any categorization method, in particular, dynamic association models such as the earth mover's distance.

The BOF approach is a nice example for media understanding of media understanding. The first cycle creates sets of descriptions. The categorization step is a second cycle in which the sets are transformed into regular histograms before they are categorized. However, the scheme could be broken up by applying the dynamic association algorithm directly on the clouds of descriptions. For example, the Hausdorff distance could be used to rate the match between two sets of descriptions. Alternatively, local descriptors could be organized by 2D strings (see first part) and made subject to rule-based classification.

That is, a number of alternatives exists for the application of local descriptions. However, computing SIFT and using BOF for categorization is a frequently employed approach. It is recommendable as a first solution for a wide range of visual media understanding problems. For optimization, though, it ap-

pears reasonable to extend/replace the standard quantization steps by problem-specific rules.

14.4 Local Description of Other Media

In this last section we investigate if the principles of local visual feature transformation are also applicable to the other media types under consideration in this book. We start with summarizing the major assets gained from the first three sections. Then we investigate if these methods can be applied to other quantitative media types. Eventually, we try to identify applications in the symbolic media domains.

We believe that local feature extraction in the visual domain provides four major tools that might be interesting for other domains as well.

- *Scale spaces* for the representation of media data at varying levels of details
- Interest *point detection* by Hesse matrix criteria
- Local *description by gradients* of neighboring samples
- Aggregation by the *bags of features* approach

Some of these methods could likewise be applied on some types of audio and biosignals. Scale space representation does not make sense in the time domain for music, speech or the detection of slow cortical potentials, because the characteristics of these signals are distributed over the entire data streams. However, the approach may be applicable in the spectral domain, where different scales could be used by divide-and-conquer algorithms. One application could be rhythm pattern detection by linear prediction, which could first be performed on coarse levels. Regions of interest could be investigated on finer levels.

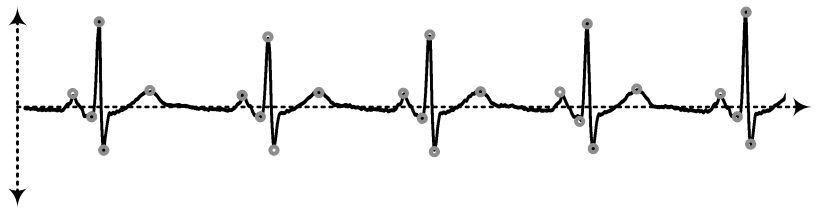


Figure 14.9: Interest Points of an ECG Signal.

Furthermore, scale spaces could be employed on all non-continuous audio and biosignal content types, including environmental sounds, P300 detection, the

recognition of spike wave complexes and K complexes. Such time-limited events require some form of *peak detection*, which could, for example be performed on coarser levels of a scale space before moving to more detailed analysis. For the recognition of such events, derivates – analogous to the Hesse matrix in the two-dimensional case – are already used today. Figure 14.9 shows a biosignal example. The peaks of an ECG signal could easily be detected on coarse levels of a scale space by taking the second derivate.

Scale spaces and interest point detection by derivates appear to be very useful for stock analysis. The first method could be employed instead of averaging, since it also creates smoothed variations of the original signal. On such a scale space, point detection could be applied for the recognition of exceptional events. Groups of interest points could be used to describe limits and patterns. We do not see a great potential for the description of audio by gradients of neighboring signals, because in these domains we are usually interested in the temporal context of an event rather than in the local structure which is anyway clear. However, in the stock domain, local description might make sense, since we are not only interested *when* something happens but also, in *which direction*. Local gradient histograms might describe interesting events appropriately. The bags of features method could generally be employed for summarization. Such a tool would be valuable for the observation of rhythmic events in biosignal understanding and possibly for the long-term observation of markets.

For the representation of symbolic media such as bioinformation and text the adaptation of the scale space approach hardly makes sense. Quantitative media lack the neighborhood relationships required for smoothing. For the same reason, point detection and local description are not translatable to this domain. The bags of features method, however, which originates from the text domain (bags of words) could as well be applied on some problems of bioinformation understanding. In particular, global and local alignment could be solved iteratively by defining bags of gene sequences and comparing bags of two genomes by, for example, the earth mover's distance. This process could be iterated from bags covering longer strings to bags of short sequences.

In conclusion, local feature extraction is state-of-the-art for object description in visual media objects. The many advantages of the approach include efficient handling of occlusions, noise and missing data, the good performance of the bags of features approach but as well a positive effect on the semantic gap and a significant reduction of the dimensionality problem. The semantic gap is reduced by local descriptions, because they imitate human visual perception and tend to cover concepts considered semantically relevant by human beings. The dimensionality reduction is reached by the few parameters of the method and the aggregation in the bags of features approach.

Chapter 15

Description of Motion

Introduces motion as a fundamental aspect of video, discusses simple motion features, the computation of optical flow and advanced motion descriptions such as camera motion and motion trajectories.

15.1 Simple Motion Descriptions

The description of motion is a detail problem of media understanding since it is not relevant to all media types, not even to all quantitative media types, but just to video. Only video has a time dimension that is able to capture the motion of subjects and objects. The special case of EMG will be discussed below. Still, we endeavor in this chapter not just to describe the principles of motion feature extraction but also to set these methods in context with general feature extraction in media understanding. As we will see, motion feature extraction is indeed related to other visual methods, in particular to local feature extraction as discussed in the last chapter.

This first section introduces the general concept of motion as well as a few simple methods for motion description. The second section deals with temporal segmentation, the decomposition of a video stream in a sequence of shots. In Section 15.3, we introduce optical flow as the fundamental concept for the motion description methods explained in the last section of the chapter. As in the preceding chapters, we conclude by relating the introduced transformations to the fundamental problems of media understanding.

In this section, we first discuss the term *motion* and show how motion is handled in video compression. Then, we introduce two fundamental description

methods: *background subtraction* and *global motion activity*. Eventually, we reflect the relevance of motion description methods for other media types.

Perceived motion is the inertia of the human visual cognition system (eye, optical nerve, visual cortex, etc.) to resolve spatial shifts over time beyond a certain degree. That is, if a sufficiently rich stream of small shifts is presented we are unable to distinguish step-wise change from continuous motion. In nature, where movement is continuous down to the quantum level, any stream of shifts is fine-grained enough to represent motion. In the artificial world of digital video already 25 steps per second (to some authors, 24) are sufficient to deceive human vision. Therefore, cinema, television and related media operate with at least 25 images (frames) per second.

Obviously, such a stream of images is highly redundant in the time domain – even more than in the two spatial ones. At 25 frames per second, each image is just shown for 40ms. In such a small time span only very limited shifts are possible. For example, the author could verify in an experiment with a ballerina that the fastest foot motions humans are capable of can be captured perfectly at 250 frames per second, i.e. in images captured with a delta of 4ms there is no visual difference. Now, compare foot exercises of a ballerina to the typical velocity of average humans and it becomes clear that human motion can very well (i.e. with high redundancy) be captured at 25 frames per second. Some artificial types of movement (e.g. rocket launches, car crashes), though, require higher temporal resolution – again for the price of even higher redundancy from frame to frame.

Now, how do state-of-the-art video compression algorithms deal with this redundancy? In the spatial dimensions, typically, selected coefficients of wavelet transforms are employed to reduce the level of redundancy. Since the temporal redundancy is even higher, a fundamentally different approach is employed over time. Instead of interpretation, a localization procedure is employed where the frames are divided into *macroblocks* of fixed size which are searched in neighborhoods of their location in later frames. The best location for a macroblock is described by a *motion vector* that gives the location delta in the two spatial directions $[\delta_x, \delta_y]$.

This procedure is similar to the edge detection approach introduced in the first part of the book and to the description of interest points discussed in the last chapter. In both cases a location of interest (here, the macroblock) is in a neighborhood compared to isomorph structures by convolution and the best match is chosen by some optimization criterion. The major difference of the macroblock approach is that motion vectors stretch over time, i.e. the best match is not identified in the object where the location of interest lies but in some temporally related object. The rest of the procedure is highly similar. Below, we will see that motion vectors play a very important role in sophisticated description methods for motion.

In this introductory section, however, we would like to limit ourselves to the most simple methods of motion description. The first to discuss is *background subtraction*. Figure 15.1 illustrates the idea. Background subtraction seeks to eliminate those parts of the visual data that do not change over time. The remaining samples are highlighted. The top left and right frames are temporal neighbors in a video shot. The bottom row shows the results of background subtraction for varying thresholds.

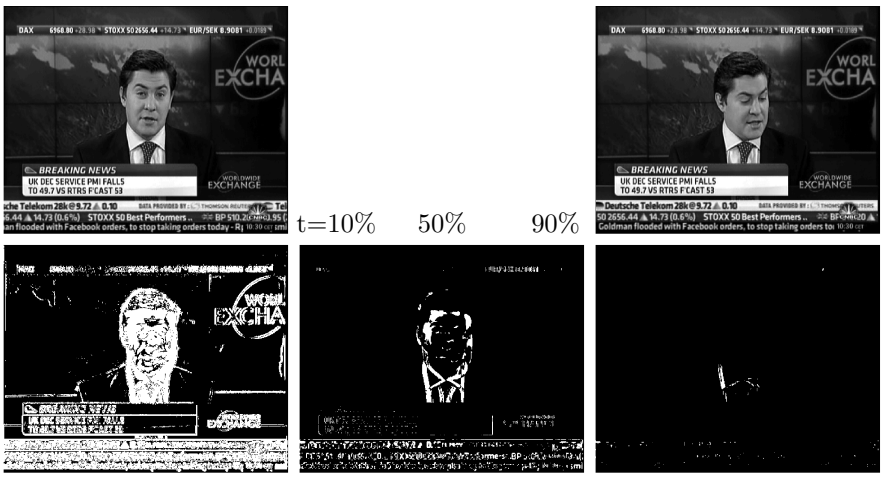


Figure 15.1: Background Subtraction for Varying Thresholds (© *CNBC*).

The figure was computed for two grayscale video frames o_1, o_2 with locations set l by the following algorithm.

```
foreach l in L(o1) do
   x(l) := | (o1(l)-o2(l) |
   if x(l) > t then
      x(l) := 1
   else
      x(l) := 0
   endif
endfor
```

Here, t is a threshold that determines the maximal difference in luminance of the two frames. If the difference is above the threshold, the sample is considered being part of motion (foreground) otherwise static background. The figure shows that the selection of the threshold is crucial for the success of background

subtraction. If it is set too small too many samples are being considered part of the motion. If it is set too high the motion vanishes.

Background subtraction is hardly ever used as a description method in its own right. It is rather a pre-processing step for the identification of spatial regions with high potential for interesting motion. It can, for example, be employed to identify the region of interest for the detection of global motion activity. In the given example, this region would certainly include the anchor person.

Motion activity is a global concept. The idea is to summarize the motion in pairs of frames independent of their sematic meaning. Several methods have been proposed for this purpose. For example, in [174] the authors of the MPEG-7 standard for media description suggest using the motion vectors of the media compression for the computation of motion activity. Lengths and directions of motion vectors can, for example, be aggregated statistically by mean and variance. If the process is repeated over rectangular image regions (e.g. a $3x3$ grid) a histogram can be computed that expresses the overall activity in a video. A different approach would use the output of background subtraction. The simplest form would count the number of non-zero samples in the output matrix x. The larger the sum the higher the activity between two frames. More sophisticated algorithms could take the position of a sample into account as well as rely on the luminance difference instead of the binary value, etc.

Motion activity can be used to get a first impression of the content of a video. It can, for example, be used to discriminate between types of content such as documentaries, romantic movies, sports, newscasts and action movies. All of these types of video have specific rates of motion activity. Another, more sophisticated application would be as a control parameter for motion compensation. Motion compensation is a pre-processing step in video object recognition. Due to the movement, object boundaries are often diffuse. Motion compensation uses the information of multiple instances of the same object to reconstruct the object boundary. The fundamental activity in a video can, for example, be used as a global estimate for the quality of object boundaries.

In the introduction we mentioned that this chapter is outstanding in the aspect that it focusses on only one media type. The majority of methods discussed here deals with the specific situation of digital video where change is encapsulated in sequences of spatial objects. However, there is one data type that also capture motion: The electromyography (EMG) is a biosignal of muscular activity, hence, describing human motion. The output of EMG capturing, however, is a sequence of potentials over time. Such a signal can be analyzed by the methods discussed in the preceding chapters. It does not contain any semantically relevant object information. Therefore, the EMG is a motion signal of a completely different type that is not relevant for discussion here.

In summary, motion is a very specific phenomenon that requires tailor-made methods. Background subtraction and motion activity provide a first idea of the

level of motion. More sophisticated approaches require the computation of dense motion vector fields as well as localization over time by temporal segmentation.

15.2 Temporal Segmentation

In this section we deal with a fundamental property of digital video. Most videos are *compositions* of multiple scenes (*shots*) that were organized in a specific semantically meaningful order by a human operator with the help of a video editing (cutting) software. The goal of temporal segmentation is the reversion of this process, i.e. the recognition of shot boundaries. This localization procedure is motivated by the hypothesis that the content of one shot is less variable than the content of multiple shots. Therefore, it should be easier to describe one shot by conventional feature transformations than an entire sequence of shots.

Temporal segmentation is one iteration in a media understanding of media understanding process. It provides the boundaries between shots. Consecutive iterations may focus on motion description within shots as well as on the description of colors, textures, objects or the content of accompanying audio tracks. Being an iteration of media understanding requires on the other hand the existence of some categorization method for segmentation. As we will see below, categorization in temporal segmentation is usually very simple. Most frequently, rule-based decision making based on static thresholds is used. Even with this simple approach high performance values can be achieved. Most shot boundaries are easy to identify.

In the remainder of this section we state a model of shot boundaries, describe approaches for the detection of edgy transitions (cuts) and of continuous transitions (fades, wipes) as well as sophisticated models that derive shot boundary information from the high-level semantics of the type of content.

Figure 15.2: Crossfade Example (© *CNBC*).

Generally, shot boundaries may take one of three forms:

- *Cut*: One scene ends with one dedicated frame and the next scene starts with the next frame. If the boundary is a cut, two shots are sequenced without overlap.

- *Fade*: The temporally first scene dissolves while the second scene fades in. the result is a spatiotemporal overlap of two scenes as illustrated in Figure 15.2.
- *Wipe*: Like in the case of a fade, the two scenes overlap. In the case of a wipe, however, a binary template (mask) defines which samples from which scene become visible when. The two scenes are mixed spatially instead of temporally. For example, it is common in sportscasts to use diamond wipes or wipes in the form of flags, logos, etc.

Cuts can be detected easily by very simple methods. For the detection of fades, adapted methods do exist that lead to a high recognition rate. Wipes are harder to detect, since many different forms do exist and the patterns of the overlapping frames can be highly variable. Luckily for media understanding, wipes are hardly used in digital video production today. Sportscasts and newscasts are among the few exceptions. Generally, the same methods can be employed for the detection of wipes as for fades, but the recognition rates are significantly lower.

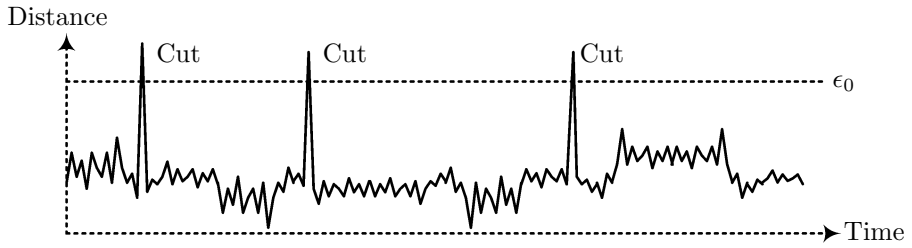


Figure 15.3: Histogram-Based Temporal Segmentation.

Figure 15.3 illustrates the general model for the detection of cuts. The vertical axis represents the differences between neighboring frames. That is, in the first step, pairs of frames[1] are compared by their properties. These properties may be aggregated background subtraction results as described above. Practically, it is often the Euclidean distance of color or luminance histograms. Whatever method is used, it is a measure for the motion activity. If the distance is high, high motion is assumed. Now, cuts are defined as points in time with exceptionally high distance, because neighboring frames belonging to different shots should have significantly different content. Hence, shot boundaries of cuts can be detected by a simple threshold ϵ_0 that defines the limit when the difference between two frames is considered beyond the possibility of being created by motion. Practical experience shows that the threshold can be set relatively

[1] Of course, we may define a hop size thus ignoring some frames between them.

easily. The difference in distance between shot boundaries and inner-shot frames is usually significant.

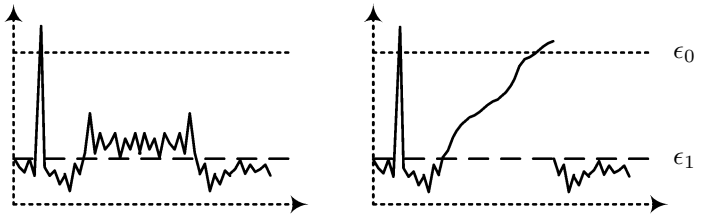


Figure 15.4: Twin Comparison Principle.

A similar approach can be employed for the detection of fades and wipes. Figure 15.4 illustrates the principle. Fades and wipes are characterized by the fact the neighboring frames that belong to different shots are mixtures of the content of these shots. In consequence, the differences between pairs of frames are smaller than in the case of wipes and cannot be detected by the threshold ϵ_0 alone. However, the distances are usually larger on shot boundaries that within the shots. The left side of Figure 15.4 illustrates this fact. The leftmost peak is a typical cut. Then, after a few frames the distance rises and stays significantly higher than before. We assume that the group of frames with the more than average distance represents a fade or a wipe. A second threshold ϵ_1 is introduced that has to be set to be above normal distances and below inter-frame distances that are typical for fades and wipes. As soon as the distance is beyond ϵ_1 it is aggregated until it returns below the threshold. Then, the sum of distances is compared to the threshold for cuts ϵ_0. If it exceeds the threshold we assume a fade/wipe shot boundary in the middle of the sequence of frames.

This approach is named *twin comparison* because it makes use of two thresholds. The aggregated sum stands for the length of the fade/wipe operation. Obviously, the recognition rate stands and falls with the careful setting of threshold ϵ_1. If it is set too low, frames with high motion within shots are falsely classified as shot boundaries. If it is set too high, relevant shot boundaries are not detected. The practical application includes the tuning of the thresholds which have to be set for each type of content (type of video, illumination, etc.) individually.

For the understanding of temporal segmentation it is important to be aware that the threshold-based methods can be employed on image descriptions of almost arbitrary content. The frames may or may not be summarized by regions, edges, histograms or moments. Their similarity can be measured as distance or by any other regression method. The resulting scores can be made subject to threshold-based quantization or some other form of binary categorization. That

is, temporal segmentation is in fact an application of visual media understanding. The output is fed back into the system as a semantically richer input.

One approach of particular importance exploits the motion vectors today usually available in compressed digital video. The motion vectors are, for example, quantized by a bags of features method and the resulting histograms are used for distance comparison. One approach could be to assume a shot boundary where medium-sized motion vectors are missing but where zero-length and very long motion vectors exist in abundance. Such data would indicate the breakdown of macroblock comparison which should usually appear at shot boundaries.

A second important add-on to temporal segmentation is to use knowledge about the production process as additional input. Such information may be fed into the categorization process that follows the distance measurement. In the simplest case, thresholds are tuned to the particularities of certain content types. In more sophisticated scenarios a ground truth can be employed to train a probabilistic model. Hidden Markov models are frequently used for temporal segmentation. This model can then be used for probabilistic inference from frame differences to belief in shot boundaries. Furthermore, an iterative process could be implemented that compares frames first on a coarse level and then likely shot boundaries on finer levels (like in a scale space).

It is not uncommon that state-of-the-art temporal segmentation algorithms reach an accuracy of 99 per cent and more. The major difficulty lies, as mentioned above, in the detection of wipes which are in many areas only of minor importance. If the same type of wipe is used repeatedly and the model is known, however, even wipes can be detected with relatively high accuracy.

Temporal segmentation is the localization step in motion description. Even for the elementary description methods discussed in the last section, background subtraction and motion activity, it makes sense to apply the algorithms on the shot level and not over shot boundaries. Furthermore, shot boundary information can be employed for aggregation, i.e. the representation of motion activity in a shot by mean and standard deviation. In the next section, we introduce a general concept for motion representation in shots which is a generalization of the motion vectors used for video compression.

15.3 Computation of Optical Flow

The key to the motion in video is the *optical flow*. The term refers to a cloud of vectors that represent the object movement between two frames. Optical flow may be computed for entire images, regions of interest, segmented objects, etc. The cloud may comprise one vector per sample, one vector per object or just one global motion vector. Optical flow may be a motion feature in its own right but it may as well be used to compute simple and sophisticated motion

descriptions. For example, motion activity can be based on optical flow as well as on background subtraction.

In this section we introduce the fundamental scheme of optical flow computation. Along the steps we explain the methods mainly used for flow computation as well as important exceptions that have been introduced in recent years. Three general approaches to optical flow computation can be distinguished.

- *Gradient-based methods* aim at identifying the most likely movement of a region of interest by *similarity measurement*.
- *Energy-based methods* compute the dislocation of an object of interest by energy minimization similar to the active contours approach.
- *Spectral methods* use the properties of certain integral transforms to compute a global optical flow.

As representative for the first group we introduce the *Lucas-Kanade approach* that combines neighborhood search with aggregation by regression. The energy-based methods are represented by the *Horn-Schunck approach* already defined in the 1980ies. Eventually, spectral methods are represented by *phase correlation*, an effective method that makes use of the properties of the Fourier spectrum.

Before we look at the details of the three groups of optical flow computation methods we would like to emphasize the general similarity between motion vectors and gradient computation. Without naming it we introduced the gradient already in the first part of the book when we discussed edge extraction by the Sobel operator. From the delta values δ_x, δ_y obtained by convoluting the Sobel matrices for horizontal and vertical over an image point and its neighborhood we can compute magnitude and orientation of the gradient as follows.

$$f_1 = \sqrt{\delta_x^2 + \delta_y^2} \tag{15.1}$$

$$f_2 = \arctan \frac{\delta_y}{\delta_x} \tag{15.2}$$

The same descriptions can be used to represent luminance gradients in the neighborhood of local interest points. The SIFT algorithm, for example, builds its point descriptors from such gradients. Motion vectors can be represented by the same scheme. As we already explained in the last section, the displacements δ_x, δ_y describe object movement over time. The computation uses neighborhood similarities like in the case of gradients. Hence, why not representing motion vectors by magnitude and orientation? In particular, the latter description element can quickly be aggregated to high-level descriptions as those described in the next section.

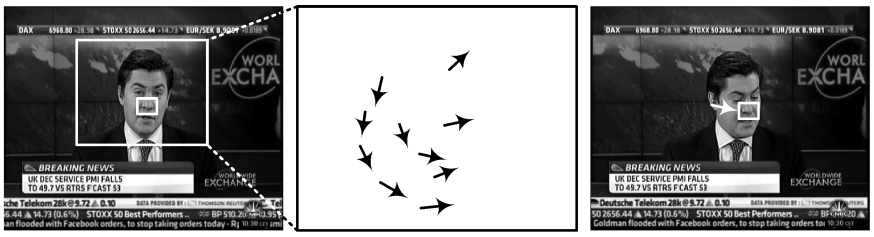


Figure 15.5: Computation of Motion Vectors (© *CNBC*).

Figure 15.5 shows a typical optical flow for a region of interest. Within the larger white rectangle we compute the displacement vectors from the left frame to the right frame (approximately two seconds later). The result includes the motion vectors depicted in the middle. The aggregation of these vectors defines the optical flow of the region of interest. As an example, the flow vector for the nose tip is emphasized. Due to head movement of the anchor person this motion vector points to the bottom right. The magnitude is small.

If we employ a gradient-based method for the computation of such optical flow, the following steps have to be taken for two video frames o_1, o_2.

1. Localization of o_1 into regions.
2. For each region do:
   (a) Extract each region as a template.
   (b) Define a search neighborhood $\theta(o_2, l_{o_1}, \epsilon)$ in the second object.
   (c) Perform convolution for each location in this neighborhood.
   (d) Select the location with the highest similarity as the most probable match for the template and compute the temporal gradient (motion vector) as described above.
3. If necessary, average neighboring motion vectors by some statistical method.

Figure 15.6 shows a typical result for two frames. As can be seen, most motion vectors are short, which indicates no or small movement. In the region of the anchor person larger movements are visible which include also some false hits, i.e. semantically incorrect matches.

In this example, the third step is only executed on the level of macroblocks and linear regression is employed for the computation of average motion vectors. This proceeding is equivalent to the Lucas-Kanade approach. It uses negative convolution (based on the city block metric) in the template matching step.

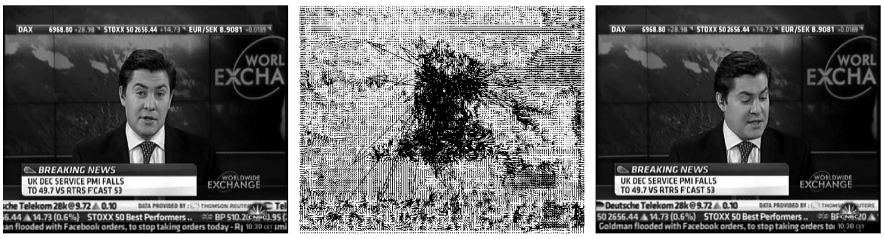


Figure 15.6: Lucas-Kanade Optical Flow Example. The figure was created using the algorithm provided by Sohaib Khan [190] (© *CNBC*).

Hence, similarity is operationalized as inverse distance. Moreover, macroblocks are used as regions.

It is important to note that in the gradient-based approach several aspects are variable. Regions may be as small as samples or as large as the entire media object. The neighborhood may be as small as a Moore neighborhood or as large as a frame. Similarity can be defined by the inner product as well as a distance measure. Aggregation may not be performed at all or even globally by computing the median of the flow vectors of all samples. Characteristic properties of the gradient-based approach are *neighborhood search* and *selection by maximal belief*.[2]

The Horn-Schunck method computes a global optical flow vector with parameterized components $f = [f_x(x,y), f_y(x,y)]$ for two frames o_1, o_2 by minimizing the following energy model.

$$\sum_{l \in L(o_1)} \left(\delta_x^l f_x + \delta_y^l f_y + \delta_t^l\right)^2 + \alpha^2 \left(\text{size}(f_x)^2 + size(f_y)^2\right) \rightarrow \min \tag{15.3}$$

Here, δ_x^l is the change of intensity in direction x at location l (equivalently defined for dimensions y, t) and α is a weight for the smoothness of the global flow vector. The energy model expresses that the variance of intensity present in the frames (first term) should be matched by the flow vector (tuned by α). The actual minimization is usually performed iteratively over all locations l individually by solving the two Lagrange equations that can be derived from the optimization criterion.

$$\delta_x^l(\delta_x^l f_x + \delta_y^l f_y + \delta_t^l) = \alpha^2(\mu_x - f_x) \tag{15.4}$$

$$\delta_y^l(\delta_x^l f_x + \delta_y^l f_y + \delta_t^l) = \alpha^2(\mu_y - f_y) \tag{15.5}$$

[2]This large set of options may be seen as an example for the curse of dimensionality.

Above, μ_x is the average for f_x in a neighborhood around l. The other variables are defined as before. The two equations show nicely the tension between intensity changes over space and time and the flow vector components. Since each location l is influenced by its neighborhood which in return consists of locations with a neighborhood that includes l, the solution for f can only be gained iteratively. That is, f is approximated over time by computing the best result for each location, re-computing the neighborhood values, and so on. A case of expectation maximization. The exact definitions of the necessary equations can, for example, be found in [208].

The last method that we would like to sketch in this section is phase correlation, a global method that employs spectral transformation. The algorithm is very simple.

1. Compute the Fourier spectra for the two given frames o_1, o_2.

2. Compute the crosscorrelation of the spectra by positive convolution: $\chi = ft(o_1).ft^*(o_2)$. Here, ft^* stands for the complex conjugate of the complex Fourier spectrum.

3. The location l of the optimal direction for the flow vector can be determined by identifying the maximum in the back-transformed correlation: $f = \arg\max_l ft^{-1}(\chi)$.

The idea behind the algorithm is that a shift over two frames (movement) where only little information is lost at the borders results in Fourier spectra that will have maximal correlation at the frequencies that remain the same. Thus, the crosscorrelation operation needs only be performed once in the spectral domain. In sample space, the operation would have to be repeated over every possible location where a shift could happen.

The presented methods are only three – frequently used – examples for optical flow algorithms. More can, for example, be found in [208]. Often, optical flow algorithms are classified as either local or global. The Lucas-Kanade would be a typical local approach. Horn-Schunck and phase correlation would be typical global approaches. However, in practice this differentiation is only of minor relevance. The latter two methods can, of course, be combined with regions of interest or other localization methods. On the other hand, gradient-based methods can be extended by aggregation or coarse representation.

Optical flow may be used as the basis for more-sophisticated descriptions (see next section) but sometimes it is also employed as a description in its own right. The author, for example, supervised experiments where Lucas-Kanade optical flow vectors where used for violence detection in videos. It turned out that statistical moments of this type of optical flow are highly expressive in this

context. In fact, the flow moments outperformed all of the state-of-the-art audio feature transformations (MFCC, LPC, etc.).

In conclusion, movement between video frames can be captured by a field of motion vectors named optical flow. Several methods exist for flow computation of which gradient-based, energy-based and spectral are three important groups. In the final section we discuss feature transformations for the description of movement-related concepts that are based on optical flow.

15.4 Flow-based Motion Descriptions

Below, we describe two fundamental types of flow-based motion descriptions: camera motion and motion trajectories. In the first case, motion is assumed to origin from movement of the camera. In the second case the origin is assumed to be object movement. Hence, camera motion describes global movement while motion trajectories describe local movement.

First, however, we would like to discuss flow-based motion activity. Above, we already mentioned that motion activity can be based on the motion vectors that are part of compressed video. In the same way, an optical flow can be used to estimate the degree of motion in a video. The fundamental problem is similar to texture description in images. Motion activity can be described in terms of coarseness, regularity and directionality. The first attribute refers to the global/local distinction. A coarse optical flow, i.e. large groups of vectors pointing into the same direction, indicates object movement while the opposite indicates camera movement. Regularity measures whether or not all flow vectors have the same gradient properties (length, direction). Eventually, the directionality can be seen as the average direction. As for textures, these three properties can be measured by statistical moments drawn from the flow field. For example, the variances of vector lengths and directions are measures for regularity while the means are measures for directionality. A motion activity descriptor could aggregate these measures over time (for example, one set per shot).

Camera motion is an important aspect of some types of video. For example, in feature films camera movement provides relevant cues on the types of scenes (e.g. action, romance). In other types of video, camera motion is of little or no relevance (e.g. documentaries). The existence of camera motion can be used to discriminate between these (and other) types of video scenes.

Camera motion detection based on optical flow is usually based on a region of interest concept. Figure 15.7 shows a typical example. Regions like c are considered highly expressive for the detection of camera motion. Regions like a are usually ignored. Regions of type b are somewhere in-between a and c. The reason is that in most videos the focus of attention is in the center of the image. Most relevant events happen there. The background is almost invisible in this

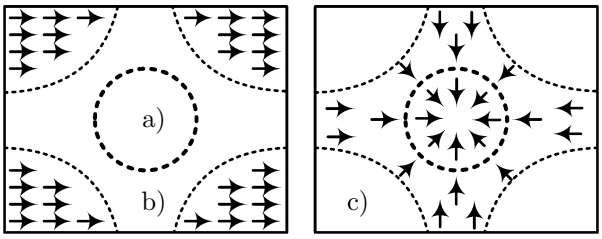


Figure 15.7: Camera Motion Detection Examples.

region. In contrast, regions of type *c* are typical background regions that can be used for the detection of camera motion.

The model of flow-based camera motion detection is very simple. We try to identify if the background moves from frame to frame in a regular way. The directionality of this movement reveals the type of camera motion. For example, the left element of Figure 15.7 shows motion vectors in the background regions that point uniformly to the right thus indicating a pan to the *opposite* direction, i.e. to the left. In the same way, zoom and other camera operations can be detected. For example, the right element of the figure shows motion vectors that point to the center region *a* which indicates a *zoom out* operation. That is, samples move towards the center of the picture because the focus of the camera is widened.

The central problem of camera motion detection from optical flow is estimating the average direction of the flow field and judging the belief in this direction. Next to the statistical methods named above, one very nice approach is based on the Hough transform (see Chapter 12). The Hough transform can be used to build a histogram of the gradient magnitudes and directions of the motion vectors in the regions of interest. The directionality can then be identified by peak detection in the histogram. Furthermore, interpreted as a density function, the neighborhood of the peak can be employed to estimate the belief in this direction. The more outstanding the peak, the higher the belief.

Motion trajectories describe object movement over time. Optical flow is the natural ground for the aggregation of motion trajectories. In fact, the flow can already be used to segment objects from their background. For example, if a group of motion vectors points constantly into the same direction we can assume that the samples/regions represented by these vectors belong to the same object. The motion trajectory describes the two-dimensional movement of such objects over time. It is, therefore, a local description that can be computed by methods similar to those used for shape description.

In its simplest form, a motion description can be the aggregate of individual motion vectors over time. Such a trajectory would be edgy and probably con-

tain a significant amount of noise. A more sophisticated approach will perform aggregation on the levels of the sample as well as the flow vectors. The first goal is achieved by the definition of regions of interest such as the before-named objects. Of course, more sophisticated object recognition methods such as template matching, edge detection or energy models can be used instead or in addition. Furthermore, the entire process can be based on interest points instead of objects. Methods such as SIFT can be used for blob detection. Aggregating over samples requires averaging the flow vectors of their composition. For this end, all statistical methods (moments, regression, etc.) can be used. The aggregation serves as a noise filter that increases the belief in the flow vectors.

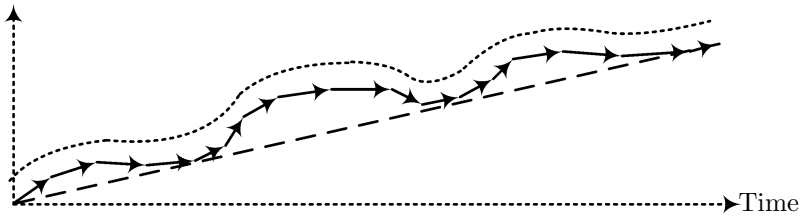


Figure 15.8: Approximation of a Motion Trajectory.

Eventually, the temporal aggregation of individual motion vectors can be improved by smoothing and averaging methods. Figure 15.8 shows an example. The arrows show a hypothetical motion trajectory which is relatively edgy. Two smoothing methods compute a simple trend line (dashed line) and a moving average (dotted line). The parameters of these approximations can be used as descriptions. Alternatively, we can try to fit a pre-defined model to the data. For example, we could define a polynomial function and adapt it to the edgy flow by an expectation maximization strategy. In the first step, the parameters of the function are guessed. Then, the gap (error) to the actual values is computed which leads to refinement of the parameters, and so on. In a similar fashion, energy-based models such as active contours could be used for template matching. In the third part of the book we will introduce the Kalman filter that is frequently used for such approximation problems.

It has to be noted that the major problem in the definition of motion trajectories is the sheer size of the data. Videos (for examples, from surveillance) can contain many objects that have to be followed over long periods of time. Computing a high-quality optical flow is already a resource-consuming task. Transforming this flow into smooth trajectories adds another costly task. Therefore, it is advisable to keep the flow computation algorithm and the approximation algorithm as simple as possible. Often, gradient-based methods without averaging are used in combination with simple statistical averaging of the trajectories.

The mass of such data can be employed to estimate the paths of walking people as well as the detection of gestures, kinematic motion, etc.

Most motion features can be made subject to categorization like all other media descriptions. For motion trajectories, however, we would like to point out the relevance of dynamic association models (discussed in the first part of the book). Measures like the Hausdorff distance fit naturally to the problem of motion trajectory comparison. Categorization can be performed by a micro process that compares the motion trajectory of interest to a reference that is associated with some human label (e.g. a particular type of gesture). In the same way, motion trajectories can be averaged in order to reduce the complexity of the categorization problem iteratively.

In conclusion, motion descriptions solve the particular problem of video description. Generally, they help to reduce the semantic gap problem, because motion is a visual cue on a relatively high semantic level. The presented feature transformations are able to extract this property considerably well. On the negative side we have the bad computational performance of motion description methods which is due to the large amount of data and the complexity of the feature transformations. Computing a smoothed optical flow is state-of-the-art in motion description and the foundation of many motion descriptions.

In Chapter 21 we will reflect the feature transformations discussed in the first and second part of this book and emphasize the current state-of-the-art methods. We will see that optical flow feature transformations are among these methods. In the next two chapters, however, we make – as in the first part – the transition from media summarization to categorization by first discussing advanced information filtering methods and then stating the categorization problem in a more general way than in the first part.

Chapter 16

Advanced Filtering Models

Lists the principal solutions for the data fusion problem, introduces several methods for feature selection, discusses methods for the smoothing of feature spaces and introduces advanced methods for redundancy elimination.

16.1 Fusion of Descriptions

This chapter continues the discussion of information filtering methods started in the first part. It serves as the transition between the block of chapters on feature transformation and the chapters on categorization of descriptions. Information filtering servers several purposes – all of which targeted at improving the *feature space* created from media content by the various feature extraction methods. The major goals of information filtering are merging of descriptions, selection of the optimal *subspace* of feature space for categorization, smoothing of feature space and redundancy elimination. In the first part, we came across redundancy elimination by factor analysis, smoothing by normalization and merging by concatenation. This chapter provides additional methods for all of these areas. Furthermore, it discusses description merging and feature selection systematically. The first and second section of the chapter deal with these two problems. Section 16.3 introduces a number of smoothing methods that help to remove noise from feature space. The last section takes up the factor analysis thread and introduces several methods that are similar to principal component analysis.

This section deals with the fusion of – possibly – heterogeneous media descriptions. In the first part we introduced simple merging of descriptions with fixed

length and suggested recipes for the transformation of variable-sized descriptions to static ones. Below, we embed this approach in a framework of description fusion methods. As we will see, merging can be performed on different levels and the selection of the appropriate method depends on the circumstances of the media understanding problem.

We first introduce the general model of description fusion, give an example for the domain of text understanding and, eventually, introduce a set of rules for choosing the right method.

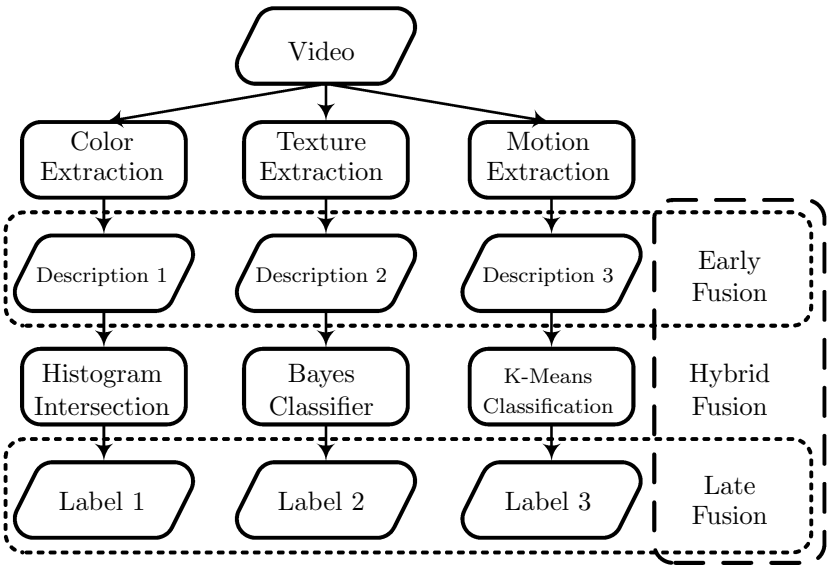


Figure 16.1: Description Merging Principles.

Figure 16.1 illustrates a media understanding process for video data where three feature transformations are used to summarize the media content. Generally, three options exist for the merging of the individual descriptions.

- *Early fusion* (also known as *feature level data fusion* and *merging*)
- *Late fusion* (also known as *decision level fusion* and *classifier fusion*)
- *Hybrid fusion*

Early fusion is performed directly after feature extraction. The merging process is usually based on concatenation of the description elements. As explained in the first part, this approach requires that the descriptions are of fixed length – a condition seldom met in video understanding, for example. If descriptions

are not per se static they have to be transformed adequately. In the first part of this textbook we suggested, for example, *statistical averaging*. Alternatively, some quantization method can be employed for coarse representation by a fixed number of description elements. Localization to a static number of windows is also an option. Aggregation methods like the bags of features approach can be used in a similar fashion as quantization methods. Eventually, in the third part of the book we will encounter dynamic filtering methods that can be used to create convergent static descriptions (e.g. the Kalman filter).

Late fusion is performed after categorization. That is, the media understanding process is performed multiply and each description is transformed into a class label by an individual classifier. This method has several advantages over early fusion. First of all, it can be applied on all types of descriptions. Their length or variability is only an issue if the classifier is limited to particular configurations. This aspect is hardly relevant in practice, since late fusion allows to select tailor-made classifiers for descriptions. A second advantage is the implicit *semantic enrichment* reached by multiple categorization processes. On the negative side stands the bad performance of late fusion processes which is due to two facts.

1. It requires, usually, more resources to train n classifiers for smaller sets of data than one classifier for a larger feature space. This drawback does not count for categorization methods that have no training step, because these usually perform complex distance measurement on the micro level which is typically of over-linear complexity. In this situation, late fusion leads to a divide-and-conquer gain in performance.

2. Individual categorization is not sufficient to clarify the semantic judgment of the media data. A second iteration is required to infer the final class label from the group of intermediate labels.

Hybrid fusion covers all approaches that mix early fusion with late fusion. This includes all situations where static descriptions are merged and all non-static parts are classified individually before their labels are added to the static feature space.

Figure 16.2 shows an example for hybrid fusion from the area of text information retrieval. A text is analyzed by three feature transformations. On one hand we analyze the structure of sentences. Furthermore we count the words and build a bags of words histogram. Eventually, we build all *trigrams* on the word level and count their frequency of appearance. The latter two descriptions deliver histograms of fixed lengths. They can, therefore, be merged on the description level without any loss of precision or generality. The sentence analysis delivers models with a complexity related to structure of the input data. It makes sense to categorize the sentences individually, for example, by hidden

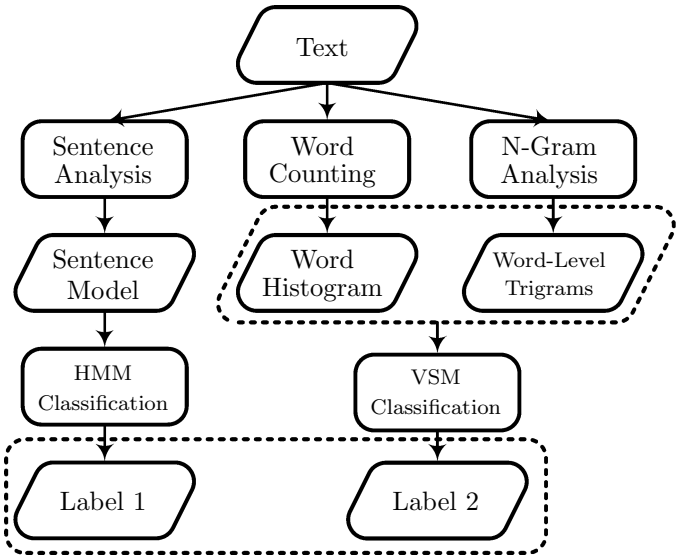


Figure 16.2: Merging of Text Descriptions Example.

Markov models. The resulting class label(s) can be merged with the result of the classification of the static description elements.

Which fusion method should be used when? Practically, if nothing stands against early fusion by merging this method is preferable from the others for the following reasons. Merged descriptions form one static feature space which can be made subject to further information filtering, in particular, redundancy elimination and smoothing. This refinement process can also be applied to descriptions separated for late fusion – but only *within* the separated data not *over* the entire space. This aspect must be considered a significant drawback of late fusion.

If the variability in the data is limited (for example, slight variations around a well-perceivable mean) it is recommendable to transform variable descriptions into static ones and perform merging for the benefits listed above. Otherwise, late fusion has to be performed, which is also the method of choice if certain parts of feature space require certain categorization methods. Furthermore, if the media understanding process is anyway embedded in an iterative refinement process it is probably preferable to perform late fusion than to sacrifice an amount of precision to averaging. Another discrimination criterion lies in the nature of the descriptions. In the first part of the book we introduced the distinction into *integral and separable stimuli* (see also Chapter 28). The first are usually made subject to distance measurement while the latter require more

sophisticated categorization. It is, therefore, reasonable to merge all integral stimuli early and to perform late fusion for the separable stimuli. In summary, the choice of method depends on the circumstances of the media understanding problem – and the experience of the experimenter.

Description fusion is a small yet important step in successful media understanding. Early and late fusion are the principal approaches. Hybrid fusion is the practical solution for multimodal applications. However we construct the feature matrix, it may be that not all description elements help the understanding process. In the next section we review methods for the selection of good description elements.

16.2 Selection of Description Elements

Before we enter categorization, we always have to answer the question whether or not all description elements help the discrimination process. No matter how clever the feature transformations have been designed, some description elements will either be *redundant* with others or even *contradict the general picture*. The selection of description elements (more frequently, *feature selection*) aims at eliminating the latter type of description elements. In this section we first introduce the general algorithm of feature selection. Then, we discuss specific algorithms that have proven successful, in particular, greedy approaches and annealing. We will see that important ingredients of feature selection algorithms are optimization criteria and measures for evaluation.

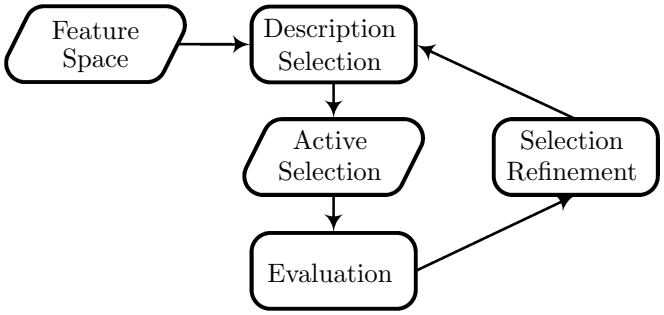


Figure 16.3: Description Element Selection Principles.

Figure 16.3 is the big picture of feature selection. The process operates on one feature space. If late fusion is used in the media understanding process, feature selection has to be performed – if necessary – once for each feature space. Naturally, the feature space will be of considerable size, since it is the purpose

of feature selection to reduce the curse of dimensionality as well as optimizing the shape of the descriptions.

The first step in the process is the selection of a subspace of feature space. Initially, this may be one description element, the entire space or some subset in-between. The selection is performed along the axis of the description elements. For each element all samples are taken. The chosen description elements form the *active selection* which is made subject to evaluation.

In the evaluation step, we judge the quality of the active selection. The definition of *quality* depends on the task. In media understanding, quality measurement usually implies performing categorization and evaluation based on the ground truth associated with the active selection. The result of such evaluation (often, the *scores*) are, for example, recall and precision values. Feature selection is also employed in other areas such as statistical approximation of data, for example, for noise elimination. Then, it is common to use statistical testing for evaluation (e.g. the t-test).

The result of evaluation is used to refine the active selection. Typical refinement operations include adding one more dimension from feature space to the active selection or removing a description element. The decision making is based on comparing the score to some pre-defined threshold. In media understanding, we could, for example, continue to add description elements to an initially empty active selection until the F_1 score of recall and precision is above 0.4 or the active selection matches the feature space. In statistics, typical thresholds are $\sqrt{2}$ or $\sqrt{\log n}$ of the t-value (n being the number of samples).

Several schemes have been proposed for feature selection, of which we would like to explain the following.

- Exhaustive search
- Forward feature selection
- Backward selection
- Selection by a genetic algorithm

Exhaustive search is a naïve approach that tries every possible combination of description elements as active selection. Obviously, exhaustive search can only be applied to very small feature spaces. Since the entire idea is irrelevant for such small spaces, exhaustive search is only a theoretical option.

Forward and backward feature selection are illustrated in Figure 16.4. The circles represent description elements. The color indicates their individual score (darker is better), which has to be computed in the initial description selection step. The left parallelograms stand for feature spaces, the right ones for the active selections. Ideally, forward selection fills the initially empty active set

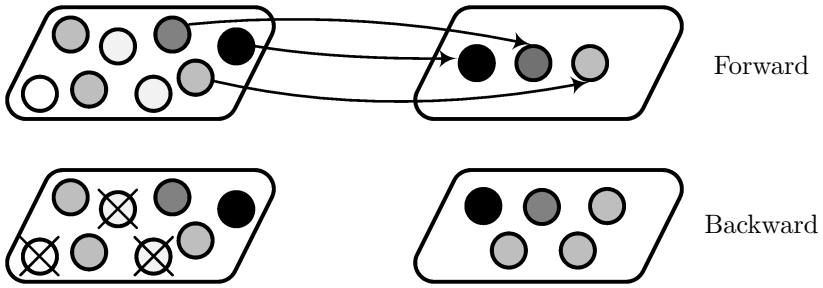


Figure 16.4: Greedy Forward and Backward Selection.

with the most relevant description elements. The process breaks off where the evaluation score exceeds the quality criterion.

Obviously, forward selection is a greedy algorithm. It will usually not assemble the best possible active selection, because joining the best individual description elements is no guarantee for an overall expressive description. One reason behind this fact is that, often, the most relevant description elements are highly redundant, while less relevant description elements add an interesting new aspect to feature space. The latter elements are often ignored by forward selection, because they do not meet the quality criterion.

Backward feature selection, no surprise, reverses the forward selection process. Initially, the active selection equals the feature space. The iterative process removes those description elements with the smallest relevance from the set until the break-off criterion is met. Backward selection is as greedy as forward selection and, therefore, has the same advantages (good performance) and drawbacks. The major difference between the two method lies in their application. Forward selection will be used where we believe that a set of description elements is highly redundant, i.e. the lot can be expressed by a few representatives. Backward selection, in contrast, is in place where the variance of the description elements is distributed more uniformly. It is employed for eliminating misleading description elements.

Forward and backward feature selection are related to cluster analysis. Forward selection can be implemented as a form of agglomerative clustering where distance measurement is replaced by score computation. Backward selection resembles divisive clustering. Elements with large distance to the others (bad evaluation scores) are removed from the data set.

Eventually, we would like to point out the importance of optimization algorithms, in particular, operations research methods for feature selection. It is obvious that the selection problem is an optimization problem ideally suited for optimization algorithms such as *simulated annealing* and other gradient-based

methods. However, the problem may be even more appealing for the application of a *genetic algorithm* (GA) because problem and solution share a number of properties. Firstly, GA are made for the evaluation of an entire gene pool which is similar to evaluating the entire active selection at once. Secondly, the crossover operator expresses a kind of respect for previously identified combinations which is similar to the accumulating role of the active selection. Eventually, the mutation operator increases the chance that otherwise neglected options enter the gene pool without a particular justification.

In Chapter 19 we discuss relevant global optimization methods including GA in detail. A practical implementation of a GA-based selection algorithm could comprise of the following components.

- The gene strings could be binary strings with length equal the number of description elements. Thus, *1* would stand for *element of the active selection*, *0* otherwise.
- Since we use a standard binary genome, standard mutation and crossover operators could be employed for breeding. Mutations could, for example, be restricted to three per cent of the number of genes per iteration.
- The evaluation function could be a process of categorization and score computation where proper classifiers and evaluation criteria are used.
- The quality criterion for breaking off the search could be combined with a maximum number of iterations for breeding in order to guarantee a minimum of performance.

It has to be noted that the property of GA that a gene pool can (and will) degenerate from a local optimum, is an important advantage of this approach. It allows for escaping from local optima and building up a better solution. The starting impetus for escape is introduced by mutation.

A number of powerful tools exist for the practical implementation of feature selection. For example, *Weka* [378] and *RapidMiner* [137] can be used to perform various forms of feature selection. It makes sense to use multiple strategies for reaching a close-optimal selection result.

In subsequent chapters we will encounter various powerful methods for evaluation that can be employed for feature selection. For example, in Chapter 20 we will introduce cross validation, a method that is state-of-the-art in greedy feature selection. In the same chapter we will discuss canonical correlation analysis which can likewise be employed on media understanding problems and on statistical problems.

In conclusion, feature selection is done for reducing the number of description elements in feature space. Goals are the reduction of redundancy and the

elimination of description elements that contradict the semantic meaning expressed in the description elements. The toolbox of methods includes all sorts of optimization algorithms. Of outstanding practical importance are the greedy search algorithms. Feature selection helps to reduce the dimensionality problem of media understanding and, in consequence, improves its performance.

16.3 Weighting of Description Elements

Before we conclude this chapter with advanced methods for redundancy elimination we would like to widen our view of normalization methods. In the first part we introduced a few approaches for global adaptation such as scaling to an interval and standardizing mean and variance. One idea that goes beyond this scheme is *smoothing of description elements*. Another is the exact opposite, *emphasizing outstanding values* in a description and suppressing others. These two principles are discussed in this section.

Ideally, we stated in the first part, a description element should have uniform distribution. Practically, this is hardly ever the case. In the visualization section we saw how odd most distributions of description elements look like. This is partially due to the failure of any sample to represent reality appropriately. On the other hand, however, imperfect feature transformations also have their share. Whatever the fundamental distribution is, the edgyness of the practical manifestations is disadvantageous for the categorization process. For example, a Gaussian Bayes classifier assumes conditional probabilities to be Gaussian. If they are not, the performance of the method lags behind the optimum. The purpose of re-weighting is to make edgy distributions of the values of description elements as smooth as possible.

The general idea is – as often – simple. Figure 16.5 illustrates t. In the first step we build a histogram of the values of the description element under consideration over all samples. Then, the histogram is compared to all relevant types of distribution and parameterization (Gaussian, uniform, etc.) – for example, by statistical testing. The closest match is chosen as the likely distribution. Eventually, smoothing is performed by first approximating weights from the differences between real and ideal distribution for the bins of the histogram and then, weighting of the values in feature space.

This very simple scheme can be replaced by a number of methods. Generally, this sort of smoothing is a template matching problem. It could, for example, be stated as an energy minimization problem and solved accordingly. Furthermore, the histogram may be interpreted as a trajectory with the implication of using comparison algorithms from this domain, etc.

The second issue is emphasizing certain values of a description element and blankening others. The reasons to do that can be manifold. For example, hu-

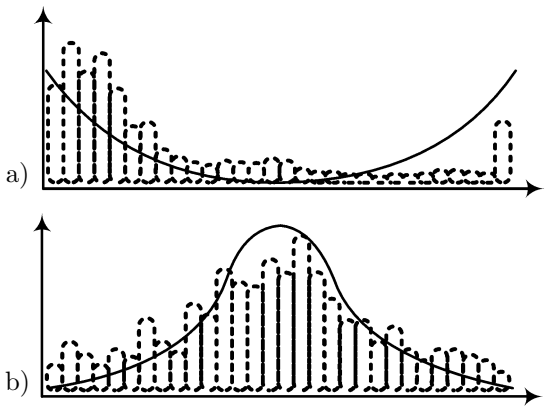


Figure 16.5: Description Weighting Examples.

man perception is in many aspects mostly attracted by outstanding values (high curvature, peaks, etc.) which justifies emphasizing such values. In his publication [275], Murdock points out that emphasizing some values on a scale for the sacrifice of others is a general property of human understanding. He calls this property *distinctiveness* and argues for its fundamental difference from similarity perception. He concludes, that both aspects – distinctiveness and similarity – influence human perception mechanisms. It is, therefore, advisable to consider distinctiveness in media understanding.

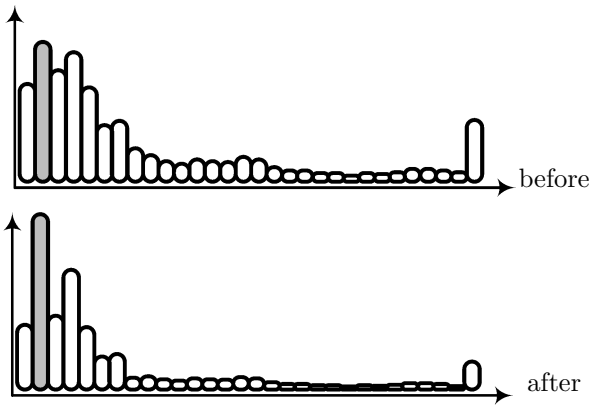


Figure 16.6: Distinctiveness Measure Example.

Murdock defines a measure for the optimization of distinctiveness for a de-

scription element f in a human-like manner as follows.

$$\bar{f}_i = \frac{\sum_{j \neq i} |f_i - f_j|}{\sum_k f_k} \tag{16.1}$$

The iterators j, k go over all samples. That is, the similarity of the value under consideration is weighted on the basis of its difference to all other values. Figure 16.6 shows the result. High values are increased, low values are decreased. The overall distinctiveness is enlarged.

The distinctiveness concept is related to a number of other ideas. If we assume the description element f to be a probability distribution, the effect of the measure becomes very similar to the *Kullback Leibler divergence* (Q5). Generally, the measure may be seen as an *interestingness measure*. Since these measures are related to evaluation measures, we will discuss them in Chapter 20.

We conclude that for the benefit of feature space a number of different normalization and weighting procedures can be applied. If it helps the categorization process we can smooth the data. If more discrimination is required we can also perform the opposite operation. Both operations may introduce new levels of redundancy in the data. Therefore, in the last section we return to the fundamental problem of redundancy elimination.

16.4 Advanced Redundancy Elimination

This section continues the discussion started in the first part of the book. Starting at factor analysis we introduce a handful of similar methods, including singular value decomposition, independent component analysis and the Isomap approach. Eventually, we discuss the outcome of redundancy elimination in the light of categorization. Is the maximization of variance in few variables really the best option for media understanding?

The main difference of this section to the methods introduced in Section 16.2 is that selection is based on ground truth while redundancy elimination ignores class memberships. Here, we use statistical approaches for the redistribution of variance in feature space. Semantic aspects are not considered – which can be seen as an advantage or a disadvantage, depending on the point of view.

Singular Value Decomposition (SVD) is a redundancy elimination technique closely related to factor analysis. The hypothesis is the same: We believe that the present *variables* (description elements) are linear combinations of a smaller set of *factors*. We aim at extracting the yet unknown factors and replacing the variables by them. The hypothesis can be stated as follows.

$$F = X.\Sigma.Y' \tag{16.2}$$

F is the feature space, X, Y are two matrices that contain in the columns the *left and right singular vectors*. Σ is a diagonal matrix of singular values. Please note that, generally, Y' would stand for the adjoint matrix of Y, i.e. the matrix gained by transposing Y and computing the complex conjugate of each entry. Since we usually do not have complex numbers in feature spaces, we can neglect the second step.

The singular values can be interpreted as the factors while the matrices X, Y provide the link to the original data. As in principal component analysis, not all factors will be used to describe the data. We will rather use only the strongest singular values as an approximation of F.

SVD is usually performed by a heuristic. In the first step, F is transformed into a bidiagonal matrix. That is a matrix where the main diagonal and the next diagonal above are filled. This operation can be performed using Householder reflection. From the bidiagonal matrix, Σ, X, Y can iteratively be approximated by expectation maximization. Guessed values are compared to F and the error is used to refine the model. The approximation can be broken off as soon as the global error sinks below a pre-defined threshold.

Singular value decomposition is closely related to principal component analysis. If F is quadratic (which is hardly ever the case in media understanding), the singular values are squared Eigenvalues. The major advantage of singular value decomposition over principal component analysis is the used approximation, which can be computed more efficiently than solving the Eigenvector problem.

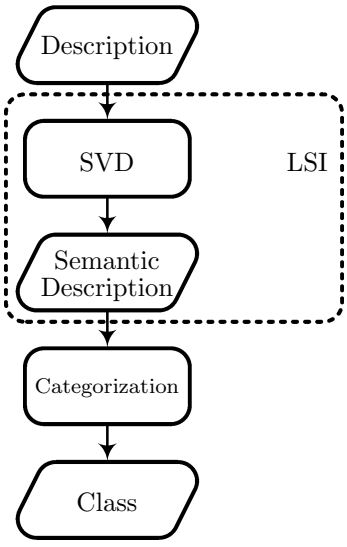


Figure 16.7: Latent Semantic Indexing by Singular Value Decomposition.

Figure 16.7 shows a typical application of the singular value composition. Latent semantic indexing is, for example, used in text understanding. The factorization is used to create so-called semantic descriptions which are then categorized into classes. Of course, the argumentation that redundancy elimination alone would lift low-level descriptions such as n-grams to a semantically higher level is rather adventurous. Being no longer able to understand and interpret the descriptions does not mean that they have a more specific context – rather the opposite. Still, latent semantic indexing is used frequently for text understanding with acceptable results.

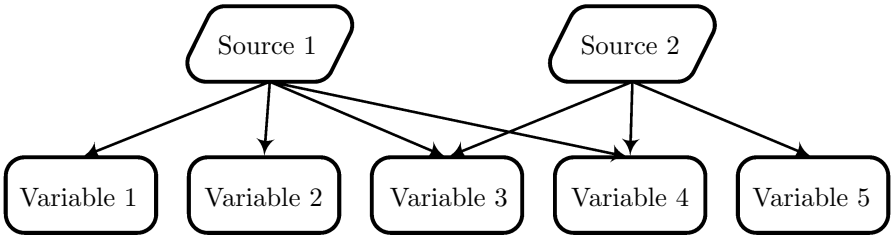


Figure 16.8: Independent Component Analysis Hypothesis.

The second approach to consider here is *Independent Component Analysis* (ICA), a blind *source separation* method. We encountered already the cross-spectral density, a source separation tool, in Section 13.2. Source separation assumes a signal to be a mixture of components. The goal of the operation is the differentiation of the components. Source separation is called *blind* if – like factor analysis – it does not make use of additional knowledge. This model is, as can be seen from Figure 16.8 identical to the factor analysis model. The only difference is that the factors are called sources in ICA.

ICA works on the vector level. For a given description f, the general approach can be formulated as follows.

$$f = X.\lambda + N(0, \sigma) \tag{16.3}$$

Here, λ is a vector of sources (factors) while X contains the weights that define the linear relationship between sources and signal. The second term is an optional noise component. The goal of ICA is to identify X, λ that explain f well and keep the components of λ maximally separated. The typical solution uses – like in principal component analysis – Eigenvectors and Eigenvalues to guess the unknown components as well as expectation maximization to approximate a good solution.

ICA is often used in biosignal understanding. For example, ICA can be employed to approximate the components of an EEG signal. In the first part

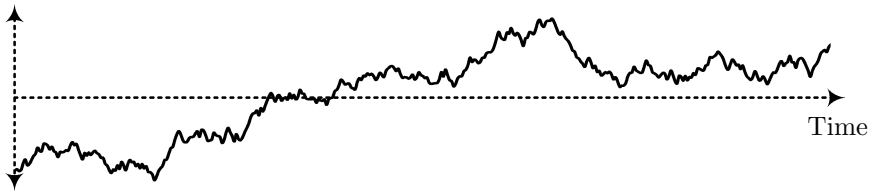


Figure 16.9: Wiener Process Example.

we emphasized that one electrode covers a large number of neurons. ICA can at least be used to difference the input coming from different regions. Furthermore, it can be used to eliminate noise components such as breath. In stock analysis, ICA is used to separate systematic from unsystematic sources. Figure 16.9 shows a typical stock signal. The fundamental components of this Wiener process can be separated into a low-frequency ascend in the first part, stagnation in the second (systematic) part and a high-level noise component (unsystematic).

Subspace analysis is a special form of ICA where a signal is assumed to have two components: a stationary component and a non-stationary component. Often, the relationship between the two components is assumed constant and linear, which makes the source separation simpler. Subspace analysis is, for example, used in EEG analysis.

The last redundancy elimination approach that we would like to explain is the *Isomap algorithm* as defined in [369]. This algorithm does not define a new model for redundancy elimination. It is based on principal component analysis. Instead, it takes the morphology of a data set into account in the analysis process. Figure 16.10 illustrates the idea. The data points form a so-called Swiss roll which is – if the structure is recognized – highly redundant. The line shows a factorization that expresses the same information as the data points at significantly lower redundancy.

The Isomap algorithm is able to compute the factors of the illustrated data set in a four-step process.

1. Build a graph of the data points that connects two data points if they are close to each other or mutual members of the set of the n nearest neighbors. The authors of [369] suggest to use one of these two strategies to build the graph. However, a mixture of both strategies leads also to interesting results.

2. Weight each edge in the graph by the distance of its two end points. Any distance function can be employed for this purpose. Of course, if the data set – like a feature space – is related to human perception, it is advisable to rely on a similarity model derived from human similarity understanding.

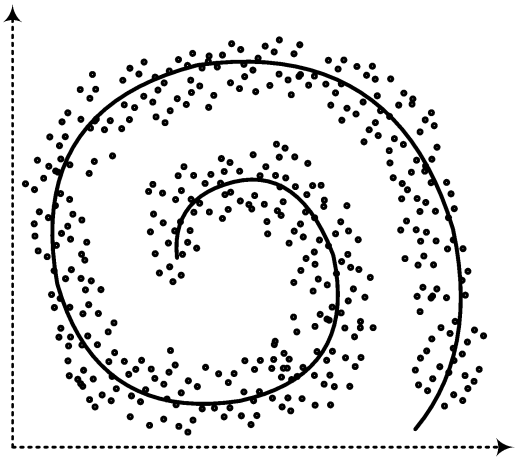


Figure 16.10: Isomap Algorithm Example.

3. For each pair of points in the graph compute the minimum distance. The minimum distance may be the direct connection or a detour over connected nodes.

4. Compute a factor analysis for the resulting graph and use the transformed Eigenvectors as representatives for the data set.

The first step enables the approach to describe arbitrary data. *Neighborhood and structure* are exclusively described by *similarity/distance*. Obviously, the third step is the major weakness of the approach. Computing the minimum distance between two points is already a non-trivial problem. The Isomap algorithm, however, requires performing this operation for each two connected points which makes it computationally highly expensive. Still, the Isomap is an interesting approach for redundancy elimination in structured data.

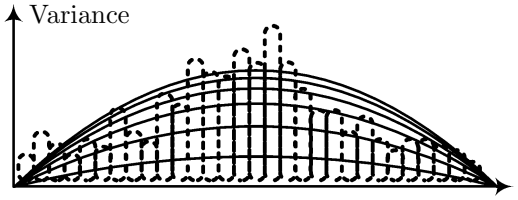


Figure 16.11: Principle of Factor Analysis.

We would like to close this chapter with reflecting the significance of re-

dundancy elimination for media understanding. Very generally, Figure 16.11 illustrates its effect. Edgy data is transformed into a smooth view that can be expressed by a small number of values. This way, a significant amount of redundancy is eliminated and the remaining variance is loaded into few highly expressive values.

We have emphasized the advantages of this process in the first part and in this chapter. Efficient data representation and dimensionality reduction are the outstanding advantages. However, redundancy elimination also has a major disadvantage. The high-frequency (edgy) components of the feature space may express important aspects of the media data as well as they stand for noise. Human perception is able to focus on fragile stimuli and to recognize their potential importance. If we eliminate the high-frequency components we run the risk of loosing the most significant source of information.

Practically, this leaves us in a trap with the curse of dimensionality on one side and the semantic gap on the other. It appear advisable, once again, to apply the media understanding process iteratively and make use of redundancy elimination in the first rounds. When the results have been optimized, it may be worth trying to run it without redundancy elimination, analyze the results and decide on the inclusion of redundancy elimination in the media understanding process based on these findings.

In conclusion, the various methods of filtering introduced in this chapter are designed to improve the performance of media understanding by reducing the dimensionality problem. By the way, noise should be eliminated as well, and it is desired that the semantic value of the descriptions should not be touched by the filtering operations. Whatever the outcome is in practice, the next step in the media understanding scheme is categorization. The next chapter reflects the general problem while the two subsequent chapters introduce a number of state-of-the-art methods for the efficient assignment of class labels.

Chapter 17

Principles of Learning Machines

Discusses fundamentals of human learning, relates machine learning to the psychological and philosophical insights and discusses selected problems of the micro process and the macro process of categorization.

17.1 Introduction to Learning Theory

This chapter continues the introduction to machine learning started in the first section of Chapter 8 of the first part. There, we already outlined the fundamental model of the *categorization process*, discussed the related terms *class, semantics, context*, introduced *references* and *ground truth* as two fundamental types of *training data*, emphasized the importance of *evaluation* by a *test set*, organized the categorization process in a *micro process* and a *macro process*, and briefly mentioned the fundamental problem of *rigidity vs. overfitting*. In this chapter, we extend the created image by insights gained in psychological science (human learning process, this section), philosophical models for the description of classes (next section), a first discussion of human-like similarity measurement (Section 17.3), and practical issues of the application of classifiers (last section). Of the latter two sections, the first targets at the micro level of categorization while the second aims at the macro level. The goal of this chapter is to show that machine learning theory is closely connected to human learning. Most algorithms used for computational categorization have equivalents in human learning theory. The quest for the ideal machine learning algorithm may, therefore, benefit from what

we know about human (social) learning.

The present section starts with facts about human learning that lead us to an ideal learning process. This model can be seen as a template for the practically used learning algorithms. It is introduced for the benefit of better orientation in the complex learning methods that are introduced in the subsequent chapters. After the learning process, the discussion continues with the problem of generalization, which is – located on the micro level – a fundamental issue in all learning algorithms. The discussion continues with the major learning problems. Implicitly, we we stated these problems already in the first part, however, listing them allows to understand their similarities and differences better. Eventually, we discuss the fundamental learning approaches, which are determined by the available training data. In the discussion, we will see that besides the above-mentioned types of training data others exist that can be used for effective categorization.

Below, we make sure that all ingredients are provided that are required for understanding and analyzing the machine learning algorithms introduced in the next two chapters as well as for the deconstruction of the general categorization problem into a set of building blocks. In this process, we will, for example, review the differentiation into *hedgers* and *separators* – introduced in Chapter 11 and extend it by the insights gained in this chapter.

Human learning is influenced by many factors. It is proven that age, sex and personal interests influence the success of learning efforts. The – sometimes, unconscious – seriousness of the effort has a large impact on our learning behavior. So have feelings and emotions. Sympathy, for example, if felt for an instructor leads to better imitation of the desired behavior, etc. There are many learning theories, starting from simple behaviorism to social constructivism and related post-modern theories. The essential factors in learning are *repetition* and the usage of *multi-medial* input. The practical importance of repetition in learning – which nobody will doubt – was one reason for abandoning the *single neuron doctrine*. According to this hypothesis, one neuron would be responsible for the recognition of a complex stimulus (for example, the *grandmother neuron* for the recognition of her face). Since the 1940ies researchers have doubted the single neuron doctrine. John von Neumann was one of the first to write in *The Computer and the Brain* that, rather, distinct patterns of simultaneously firing neurons represent stimuli. Today, this view is generally accepted. Among others, it explains why multi-sensory input helps the learning process. More input allows better training of the neural spiking patterns.

Figure 17.1 illustrates an ideal learning process. Learning is generally *iterative*. Therefore, we have a link back from adaptation to the stimulus. Human learning is driven by stimuli, i.e. some patterns presented to the human senses. In the first step of learning a mental representation is constructed. For example, in visual perception the input stimulus is first disassembled into a group of lines

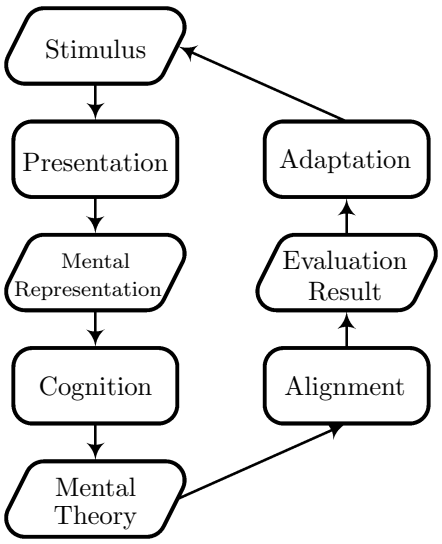


Figure 17.1: The General Learning Process.

with coarsely quantized lengths and orientations. The mental representation is then made subject to a cognition process, which results in a mental theory (for example: *This is the face of my grandmother*).

In the next step, the mental theory is compared (aligned) to experience/reality. That is, we try to confirm that the cognition process was conducted in the correct way. All learning aims at the construction of a conflict-free mental theory. The result of the evaluation can be used to adapt the stimulus for the correction of misinterpretations. The eventual result of the learning process is a mental theory of the presented stimulus.

We would like to point out that this learning process covers human learning as well as machine learning. In the latter case the left side of the figure stands for the training process, the right side for evaluation and refinement. In the training process, the mental representations are equivalent to the descriptions extracted from media object. The cognitive process is the actual categorization process.

Similarity assessment is a ubiquitous element of categorization processes. Similarity assessment allows us to recognize stimuli as similar that differ only little in their mental representation. This ability is called *generalization*. It accelerates the learning process of related stimuli significantly and it allows to map learnt concepts on unknown stimuli that are somehow comparable to known ones. The fundamental problem of generalization is to state *how similar two*

stimuli are depending on the difference (distance) of their representation. The simplest answer to this question would be the Dirac δ function which would be maximal for zero distance (the stimuli are identical) and minimal otherwise (no similarity for any form of deviation). This function has not be considered as a generalization function for human learning, even though it is relevant, wherever the elements of the mental representation are not related, i.e. *where no concept of neighborhood* exists. This is, for example, almost the case for bioinformation and perfectly the case for random strings, which explains why it is so hard for human beings to learn random sequences.

Practically relevant generalization functions are limited distributions. It is not surprising that the Gaussian distribution plays an important role in the (historic) definition of generalization functions. Applied as a function for *similarity from distance*, the normal distribution expresses that closely related mental representations are considered highly similar, while representations with high distance are considered unsimilar (see the first graph in Figure 17.2).

The Gaussian generalization function has seen considerable criticism from psychologists who investigated the actual generalization behavior of human beings. Of paramount importance are the works of Shepard, who stated in [336] that the ideal generalization function would be $e^{-m(x,y)}$, m being a distance function for objects x, y. The resulting graph is the second element of Figure 17.2. This *universal law of generalization*, as Shepard called it, was already suggested in [310]. It assigns perfect similarity only to identical representations. Any form of difference is punished, initially stronger than in the Gaussian case, for large distances to a lesser degree.

Recently, the universal law of generalization was generalized in [370]. The authors point out that humans show a certain flexibility for small differences in representations. Hence, for small distances, the similarity score should not be penalized. Please observe, that this most recent form of generalization function, the bottom element of the figure, is again, very similar to the original Gaussian function. In conclusion, generalization is an important aspect of human learning which should also be considered in machine learning, if the set of stimuli possesses a concept of neighborhood.

Which learning problems exist that can be solved by the ideal learning algorithm? Many authors agree, that three fundamental learning problems exist both for humans and machines.

- *Recognition of patterns*
- *Estimation by summarization* (also known as *regression*)
- *Estimation of a density function*

Figure 17.3 illustrates these three problems. Pattern recognition aims at associating an unknown stimulus with the nearest (smallest distance) known

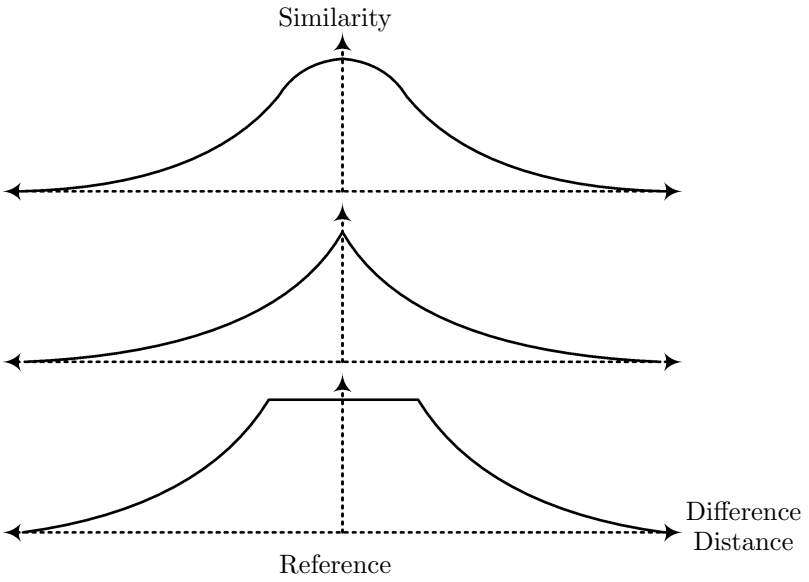


Figure 17.2: Generalization Functions: Gaussian, Shepard, Tenenbaum.

stimulus. The figure shows that the circle (unknown stimulus) is associated with the closest x. Human pattern recognition is frequently called *association*. Vocabulary learning is a typical example. Here, the assignment process is heavily influenced by our generalization behavior. An example for machine pattern recognition is face recognition.

Regression aims at the summarization of a set of stimuli (that may represent a sequence or not) by one mental theory. The figure shows a dotted trend line as a typical regression example. Human beings are very well able to perform regression visually. In machine learning, one typical application would be linear regression for prediction in stock data analysis.

The last learning problem, density estimation, requires the most sophisticated procedure. Here, we aim at learning the characteristic function of a set of stimuli. This is what we usually call the *experience* of human beings. Density functions (for example, the dashed ellipse in the figure) allow us to judge a new stimulus as, for example, dangerous, interesting, desirable or typical. It has to be noted, that human density estimation is necessarily a long-time process. Collecting experience is a learning process with many iterations and adaptations. The non-existence of stability of the learning environment makes the process even more complex. Furthermore, in the third part of this book we will discuss the major findings of *norm theory* and see, that human density functions are –

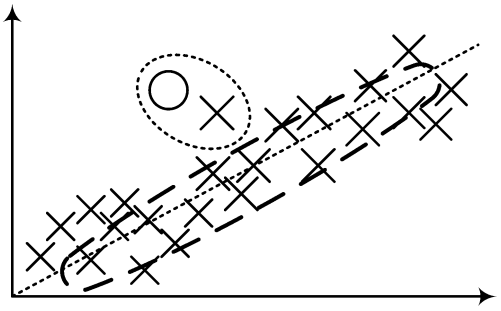


Figure 17.3: Pattern matching (dotted ellipse), Regression (dotted line) and Density Estimation (dashed ellipse).

despite many years of learning and experience – often surprisingly insufficient.

In machine learning, density estimation is the core component of all probabilistic methods. We already mentioned the typical methods used, in particular, expectation maximization and Gibbs sampling. Both procedures are iterative and resemble the general learning process as well as the process of human experience collection.

Now, how can these fundamental learning problems be solved? Scientific effort has developed three general approaches and a number of more specific ones. The three major approaches are discussed below.

- *Supervised learning*
- *Unsupervised learning*
- *Reinforcement learning*

Supervised learning is the case discussed in the first part of the book. Ground truth data and labels are used to train, evaluate and refine a classifier. Supervised learning is the ideal case illustrated in the general learning process. In human learning, the ground truth labels are personified by the teacher that represents world knowledge and enforces refinement of the mental theory.

Unsupervised learning builds a mental theory only from stimuli. The cognitive process that transforms mental representations into a theory has, in the absence of feedback/alignment, to be controlled by a set of rules. In machine learning, the k-means algorithm is a typical unsupervised learning algorithm. There, the references together with the distance function and the *The nearest reference wins* rule form the set of rules required for learning. Obviously, any unsupervised learning problem can be stated as a supervised learning problem. The opposite is not true. Hence, supervised learning is a more general solution

than unsupervised. A typical supervised algorithm is the decision tree. Decision trees are able to represent *any* ground truth.

Multiple instance learning, *transduction* and *semi-supervised learning* are three special cases of supervised learning. In the first case, we do not provide class labels for each stimulus but only for *groups* of stimuli. This proceeding allows a little more flexibility in the learning process. Transduction learning aims at estimating outputs from ground truth data and additional references. In a similar fashion, semi-supervised learning combines feedback-based learning with rule-based learning.

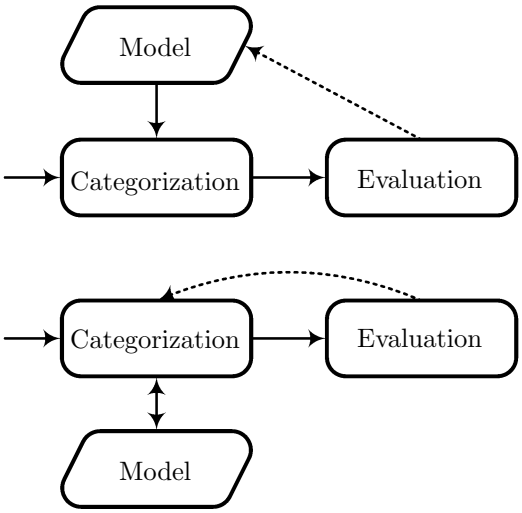


Figure 17.4: Supervised Learning (top) and Reinforcement Learning (bottom).

Reinforcement learning is a concept that looks similar to supervised learning but is, in fact, fundamentally different. The main idea is that learning is performed unsupervised but under the impression of the caused effects. The learnt patterns are evaluated as positive or negative and a corresponding *reward* is fed back into the system. Figure 17.4 illustrates the difference between supervised learning (top graph) and reinforcement learning (bottom graph). In the second case, the results of evaluation are not used to manipulate the categorization model (mental theory) directly. Instead, the rewards are made known to the categorization process which in turn manipulates the model. We admit that the theoretical difference between supervised and reinforcement learning is small. With little effort the latter concept can be incorporated into the first.

In this section we approached the categorization problem from the psychological side: learning. We came to know the paramount importance of generalization

for effective learning. That is, if the given data is not organized by some neighborhood concept, learning will necessarily be ineffective. The three fundamental problems of learning can be approach by supervised or unsupervised learning. The first concept, more general than the second, is the typical model in media understanding. In the next section we take the philosophical approach by trying to answer the following naïve question: What is a class (semantic group, mental theory, category, etc.)?

17.2 Concept Theories

The result of categorization is a label that is associated with some *meaning*. In computer science, we frequently call this meaning a *class*, in psychology a *category*, in philosophy a *concept*. Philosophers have discussed – and are still discussing – the nature and structure of concepts for more than 2500 years. This discussion has lead to a number of *concept theories*. We believe that the philosophical understanding of concepts is able to provide valuable input for understanding the machine learning problem. In this section, we review the five major concept theories and try to link them to the properties of categorization methods. We follow the excellent introduction given in [245].

We require a few terms for understanding the concept theories. Concept theorists differentiate between *primitive concepts* and *complex concepts*. The first are close equivalent to our *descriptions*. The second type may be seen as *events* (high-level semantics). In Chapter 28 we will see that the two types of events can also be distinguished by *surface features* and *deep features*. Primitive concepts are characterized by clearly recognizable surface features while complex concepts are characterized by hidden (deep) properties. Below, we will see that the existence of deep features is a central problem of the definition of concepts.

In [245], Rosch defines the terms *category*, *prototype* and *taxonomy*. A concept is a set of equivalent objects. Of these, the prototype is the most representative one. A taxonomy relates categories to each other by their similarity. These definitions are very useable for the discussion below.

The five major concept theories are:

- Classical theory
- Neoclassical theory
- Prototype theory
- Theory theory
- Conceptual atomism

The classical theory of concepts is by far the oldest as it was already discussed by the classic Greeks. Prototype theory is a product of the early twentieth century while the three other theories were developed in the intensifying discussion process of the 1960ies.

The classical theory states that any concept can be defined by a set of *necessary* and *sufficient conditions*. The classical theory tries to get a hold on reality by tools of mathematical rigor. For example, a dog is an animal with a snout, a tail, fur and claws that barks, eats meat, nibbles at bones, etc. All of these conditions are necessary, but they are all together not sufficient – which brings us to the first major problem of the classical theory. In its long history it was hardly ever possible to define a practical category by such conditions. This is called *Plato's problem*. Of course, a theory – however clearly defined – that does not solve the concept problem practically, will hardly be satisfactory.

Other problems of the classical theory are summarized in Table 17.1. It is not able to explain prototypes, i.e. why one member of the set of a category should by more relevant than another (*typicality problem*). It cannot explain why humans are able to categorize an event correctly even though they are ignorant of or wrong about the necessary and sufficient conditions of the event. Moreover, psychological experiments show that necessary conditions appear to be irrelevant in the cognitive categorization process. As with typicality, the classical theory is not able to model fuzzy boundaries of concepts. Eventually, from a postmodern point of view classical theory is prone to construct concepts rather than to describe them (*analyticity problem*).

The neoclassical theory tries to overcome the major problems of the classical theory by relaxing the model. The need for sufficient conditions is dropped and it is no longer claimed that the theory should work for all concepts. Those that can be made subject to the neoclassical theory are described by an as accurate as possible list of necessary conditions (or, properties).

The prototype theory implements a completely different idea. It states that the core element of any concept is its most representative example. Objects that are sufficiently similar to the prototype (generalizable) are assumed to be members of the category. This theory is very appealing for media understanding since it uses the same tools. For example, a dog can be seen as anything sufficiently similar to a German shepherd. Unfortunately, it has been shown that prototype theory suffers from significant shortcomings. The theory is hardly able to express composed concepts such as *old female dog belonging to an Indian shipbuilder* by a prototype. As the classical theory, it works well despite of error and ignorance of the prototype. Plato's problem is – to a lesser degree, but still – true for the prototype theory and it is difficult to find an undisputed prototype for such a simple concept as *car*.

Figure 17.5 compares classical and prototype theory. The gray, diffuse concept is, in the first case, fenced off by a number of border lines (conditions).

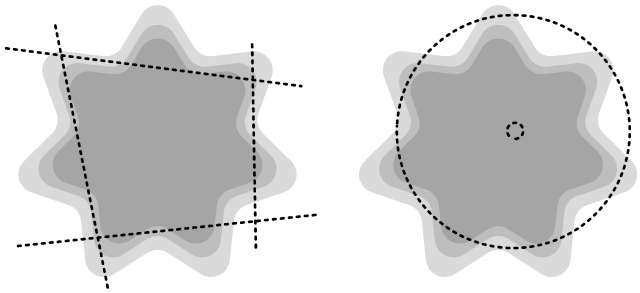


Figure 17.5: Classic Concept Theory (left) and Prototype Theory (right).

In the second case, we define a center and a radius for everything that should belong to the concept. As the figure shows, both methods fail in describing the concepts perfectly. Every part of the concept outside the concept border may be interpreted as underfitting (too rigid), every close adaptation of the border to the concept as overfitting.

Theory theory and *conceptual atomism* are two approaches of the recent past. The first theory states that the *mental theory* is work in progress and a mixture of properties and examples that cannot be made subject to analysis. In view of the classical and prototype theory this approach appears reasonable. However, it does not help us much (impotence problem), like the others it cannot explain cognitive errors and ignorance, and it does not provide stable definitions of concepts. It rather leaves the recognition to the life-long learning process.

Conceptual atomism is even more radical. This theory assumes all concepts to have no structure and, therefore, to be not analyzable. Though this approach removes the major problems of the other approaches, it opens significant new ones. For example, some concepts, such as *door*, obviously have a structure and can be analyzed. A door consists at least of a door leaf and a handle. Atomism cannot explain composed concepts at all and – very important for us – it does not explain much. It is tempting, though, to combine conceptual atomism and the neoclassical theory, defining some concepts as atoms and the rest by atoms and necessary conditions.

Table 17.1 summarizes the shortcomings of the five concept theories. What can we learn from these theories?

1. There are not so many possibilities to define classes. 2500 years of philosophical investigation have yielded only two analytic theories.

2. The classical theory is very similar to those classifiers that we called *separators* in Chapter 11. Here and there we separate relevant and irrelevant events by conditions.

Aspect	*Classic*	*Prototype*	*Theory*	*Atomism*
Analyticity	+			+
Compositionality		+		+
Error	+	+	+	
Fuzziness	+			
Ignorance	+	+	+	
Impotence			+	+
Irrelevance	+			
Plato's Problem	+	+		
Stability			+	
Typicality	+	+		

Table 17.1: Comparison of Concept Theories.

3. The prototype theory is very similar to what we called *hedgers*. Both approaches define a middle point (reference, prototype) and assume a neighborhood of the typical element as relevant.

4. Since all efforts to define rational concept theories have lead to just two theories, which are both equivalent to fundamental types of classifiers, we can conclude that hedgers and separators are the fundamental approaches to machine categorization and that no third fundamental possibility exists.

5. The best way to describe concepts is probably a mixture of the theories. Hence, the best machine categorization scheme may also be a mixture of separation and hedging. In machine learning, ensemble methods (e.g. boosting) implement this idea (see Chapter 19).

In the two remaining sections we deal with details of categorization on the micro level and the macro level that are relevant for the next two chapters where important practical classifiers are introduced.

17.3 Similarity Measures in Categorization

The author of [90] is more strict than we are. He distinguishes *classification* from *categorization* by requesting from the latter that the category must express the properties of the classified object. Consequently, he distinguishes arbitrary *classes* from *categories*, of which the latter can be described by the methodology of a concept theory. However, he arrives at the same conclusion as we, when he states that *similarity is at the heart of classification.* Here, classification is the

macro process while similarity measurement is the micro process embedded in and iterated by the macro process.

In this section, we deal with similarity measurement methods that extend the already introduced metric distances and generalization models. We already discussed the shortcomings of distance models that operate on *dimensional* descriptions [90]. Psychologists have found that *on/off-features* (predicates) allow more freedom in the definition of human-like similarity measures. Appendix B.2 summarizes such measures. Below, we briefly discuss the transition from dimensional distance measures to predicate-based measures as well as the application of these measures on categories described by sets of predicates. En passant we introduce a model for human choice behavior – a problem closely related to similarity measurement.

In the first part of the book we introduced the vector space model and stated that a number of significant distance measures are based on the metric axioms. We discussed the Minkowski distances of first and second order. Another distance family of high significance are the Mahalanobis distances. Equation Q4 in Appendix B.1 shows one particular form. The general model is defined as follows.

$$m(x, y) = x.\chi.y \tag{17.1}$$

Here, χ is a covariance matrix of the elements of the description vectors x, y. By setting χ we are able to express relationships between groups of description elements flexibly. Therefore, Mahalanobis distances are used wherever such covariances do exist (for example, between colors in color histograms).

The metric axioms, however, are too rigid to model human similarity judgment. We saw that the symmetry axiom as well as the triangle inequality are violated by human judgment. Among the remedies we have the density model developed by Krumhansl (M1 in Table B.3). This model adds a density term for each of the two compared stimuli. The idea is that smaller distances are more relevant in high-density areas than in areas with lower density. The distance terms create this effect and are thus able to overcome the restrictions of the symmetry axiom and the triangle inequality.

Since the Krumhansl model makes use of a distance function, it is a metamodel of similarity measurement. A second model of interest here is the product rule defined by Estes (M3 in Appendix B.3). This model defines a static distance m_i for non-identical description elements and computes the overall distance as the n-th power of this distance, where n is the number of non-identical description elements. The practical value of the product rule is limited, because, as the author himself states, *the product rule may be excessively sensitive*. However, as a means to overcome the limitations of the metric axioms it represents an original approach.

The categorization process of media understanding may also be seen as a problem of *choice*. Given a media event, the categorization algorithms has to choose the best fitting class label. Choice models make use of distance functions. For example, the model of human choice introduced by Luce [238] defines the probability $P(c_i|f)$ that a particular category c_i is associated with a description f as follows.

$$P(c_i|f) = \frac{m(c_i, f)}{\sum_j m(c_j, f)} \tag{17.2}$$

For the approach, the c_i are considered prototypes (references) of their categories. Hence, Luce developed his model after the prototype theory. This model was later extended by Shepard, who converted the distance measurement process into a similarity measurement process by using his generalization function $e^{-m(x,y)}$. In both forms, the model states that the likelihood of a description belonging to a particular category depends on the relative proximity of the description to the category prototype. It would be interesting to incorporate this form of weighting, for example, in the distance-based models introduced in the first part. Furthermore, it would be interesting to investigate the performance of a choice model that uses both Shepard's generalization function and Krumhansl's density model. To the author's knowledge such experiments have not yet been undertaken in media understanding.

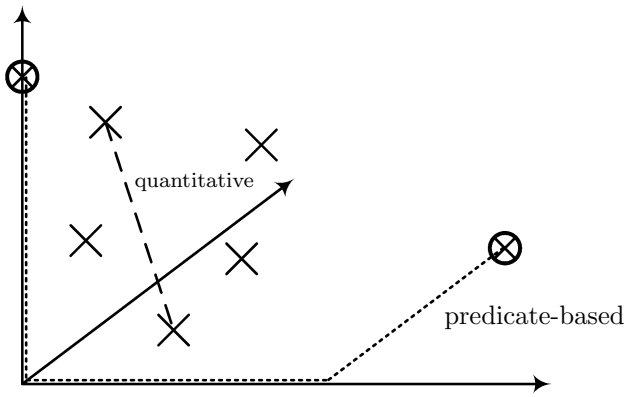


Figure 17.6: Quantitative and Predicate-Based Similarity Measurement.

Psychologists suggest a general approach to overcome the problems of dimensional similarity measurement. Predicate-based measurement allows for defining models (in fact, meta-models) that are not metric. The fundamental idea is illustrated in Figure 17.6. Dimensional descriptions are organized along quantitative

scales. Distance can be measured along arbitrary paths. Predicates, in contrast, are properties with only two values: present (on) or not present (off). Hence the frequent name *on/off-features*. Since predicates carry only little information, predicate-based descriptions must necessarily be longer than dimensional descriptions. The comparison process is simpler, though. Two corresponding predicates can either be both present, both absent or only one of the two carries the property. The table in Appendix B.2 lists a large number of predicate-based similarity and distance measures. These models are meta-models, because the property-wise comparison is already a similarity measurement process.

Some predicate-based measures have proven exceptionally successful in the representation of human similarity perception. Tversky's feature contrast model P6 is able to express most findings about human categorization. The Hamming distance P3 is successfully employed in text understanding. The pattern difference P8 is a successful measure in cluster analysis. However, one major problem arises when dimensional distance measures should be substituted by predicate-based measures in media understanding. Media description elements are usually not predicates but quantities. A straightforward solution to this problem is designing a media understanding of media understanding process that computes predicates from quantities in the first iteration and employs predicate-based similarity measures in subsequent iterations.

Findings in psychological research suggest a second solution. Recent experiments have shown that both distance measurement of quantities and predicate-based similarity provide valuable information for the categorization process. Authors have suggested using both types of information in *dual process models*. That is, where quantities are provided by feature extraction, quantitative models are used for comparison and where predicates are provided, predicate-based measures are used. Furthermore, predicate-based measurement can be extended to quantities by replacing the inner comparison process (both on? both off? otherwise?) by fuzzy methods. The table in Appendix B.4 lists a few possibilities. We will discuss them in detail in Chapter 28. For the purpose of this section it is sufficient to understand that predicate-based and quantitative measurement are not mutually exclusive. Both approaches can be used in the micro process of categorization, individually or combined.

In summary, metric distance measurement is not the only option for the categorization micro process. Human choice models can be incorporated as well as generalization functions and other distance meta models. If distance measurement is insufficient, the process can be extended by using predicate-based measures, either as a replacement of quantitative models or combined in dual process models. We are positive that human similarity perception will gain more attention in machine learning in the near future, and that flexible dual process models will replace static distance functions.

17.4 Classifiers in Practice

This section discusses a number of relevant issues on the macro level of categorization. We extend and reorganize the terminology of the first part of the book. First, we refine the fundamental applications of categorization and relate them to the learning problems. Then, we deepen the categorization of classifiers into hedgers and separators and give examples based on the already introduced algorithms. Eventually, we emphasize a few macro level concepts important in the next chapters, and we prepare a framework of building blocks for their analysis.

In the first part, we listed the three fundamental applications of categorization methods as *matching*, *retrieval* and *browsing*. Matching aims at identifying one correspondence for a query in a database. Face identification is a typical example. Retrieval gathers the n best database members for a query. Internet search is a typical example. Browsing organizes a database into (hierarchical) clusters. Music genre classification is a typical example.

One application is missing in this list. *Prediction* aims at extrapolating semantic knowledge from a query with or without knowledge from a database. Prediction is typically an application of regression. A typical example is stock data analysis for prognosis. The two other fundamental learning problems, pattern recognition and density estimation, are only of minor interest for prediction. Density estimation would theoretically be applicable, but is practically hardly used. Instead, typical applications lie in retrieval and browsing. Pattern recognition is the characteristic learning problem for matching applications.

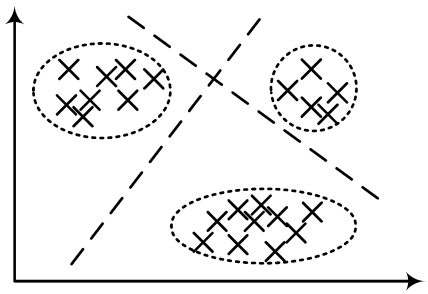


Figure 17.7: Hedging (dotted) and Separation (dashed).

In Chapter 11 we introduced a simple taxonomy for categorization methods by declaring that a classifier is either a *separator* or a *hedger*. Figure 17.7 illustrates the fundamental difference. Separators define categories by drawing a line (hyperplane) between sets of objects in feature space. Hedgers, in contrast, fence off accumulation points. It appears fair to characterize separators as *global* operators while hedgers are rather *local* operators. A separation line goes through

the entire feature space. One hedge defines just one cluster. Hence, hedgers may be seen as ideal retrieval algorithms. Browsing will rather be implemented by separators.

The construction of a local hedge requires a few tools. One option is the provision of a center point and a radius (for a circular object) or some conic section rule. Another option is the provision of a density function (mixture) that defines the limits of the hedge. Separation requires the definition of a selection rule (decision rule). Depending on their model, the already introduced categorization methods can be classified as follows.

- Hedgers: Binary Independence Model, Cluster Analysis, K-Means, K-Nearest Neighbor, Vector Space Model
- Separators: Decision tree, Random Classification

It is obvious that the methods that use references belong to the hedgers. The binary independence model is a hedger, because it relates database members to a query without defining a cut-off criterion for the result set. That is, it performs a reorganization of feature space around the query. Decision trees are typical separators. Random classification is just a theoretical option used as a baseline for the evaluation of algorithms.

The majority of the probabilistic methods is missing in this list. Generally, Bayesian classifiers and Bayesian nets (including all Markov processes) are rather separators than hedgers, because the estimated densities resemble division lines between feature space subsets. On the other hand, the adaptation of the density functions is sensitive to cluster size and location, which would be a criterion for a hedger. Therefore, we rather add them to a new group, the *intermediates*. These methods are located somewhere in-between hedgers and separators. In the consecutive chapters we will endeavor to characterize all new categorization methods as either hedgers, intermediates or separators.

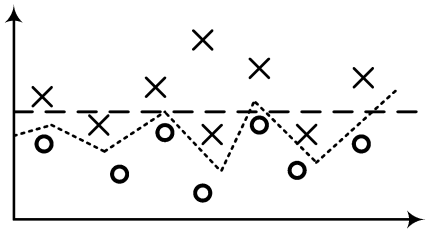


Figure 17.8: Underfitting (dashed) and Overfitting (dotted).

No matter whether they are hedgers or separators, all classifiers have to find a balance between the *rigidity of the model* and a *tendency for overfitting*.

Figure 17.8 illustrates the problem. The dashed line is a classifier that separates objects of type x from those of type o. As we can see, the rigidity of the model (the line cannot be bent) forces the algorithm to a significant amount of misclassifications. In contrast, the dotted line separates the data perfectly. The price is close adaptation to the data. It is likely that this classifier would fail for a more complex data set. This effect is called overfitting.

For the taxonomy defined above we can say that, practically, separators are more extreme in terms of rigidity and overfitting. Too simple decision sets are often too rigid while too flexible ones tend to be overfitting. Experiments in Weka show that decision tree methods (e.g. random forests) frequently outperform all other categorization methods. However, the performance is often due to overfitting to imperfect ground truth. The behavior of hedgers depends on the references. If they are well chosen, good classification results can be achieved. Here, *well chosen* may be criticized as an open door for overfitting. In order to overcome this criticism it is important to use hedgers that define/refine the reference points based on some general model, not user input.

The last sentence leads the discussion towards the *complexity* of categorization algorithms. Decision trees are very simple algorithms that can be trained and executed quickly. Adaptive hedging procedures based on the k-means algorithm require significantly more training time. For practical application it is important to balance the importance of good performance (often, linked to rigid models) against the generalization potential of the solution. It is generally advisable, to start the search process for the ideal classifier by comparing the performance of the standard algorithms for the feature space, analyzing the complexity of the model of the best performers and using the algorithm that has the best relationship of *performance vs. model complexity*.

We would like to close this chapter with a first sketch of potential *building blocks* of categorization. In Chapter 21 we will develop a detailed model. The majority of the algorithms introduced in the first part of the book performed either *similarity measurement* on the micro level (cluster analysis, k-means, etc.) or *probabilistic inference* based on *density estimation* (Bayesian methods). Some methods perform *quantization* as a preparatory step or for simplification on the micro level (e.g. decision trees, k-nearest neighbor, binary independence model). The macro process of most algorithms loops over a data set in at least two cycles (training, application), densities are computed by iterative processes and categorization results are refined iteratively. Hence, *control loops* are characteristic elements of categorization processes. In summary, four major building blocks of categorization can be summarized as:

1. Control loop (expectation maximization algorithm, threshold optimization, training cycle, iterative refinement by relevance feedback, etc.)

2. Model estimation (selection of weights, density estimation, probabilistic inference, etc.)

3. Quantization (thresholding, selection of description elements, decision rule application, etc.)

4. Similarity measurement (distance measurement, generalization, choice, etc.)

Some building blocks are linked to particular types of classifiers. For example, model estimation is a necessary element of probabilistic models. Hedgers will usually make use of similarity measurement. Simple algorithms make no use of control loops. Quantization is the only building block that is as important in categorization as in feature transformation and filtering. The simplification and generalization of data is a fundamental method of media understanding.

We conclude from this chapter that categorization is the fundamental step in the media understanding scheme for raising the semantic level of the data. It has, therefore, a positive influence on the semantic gap and the polysemy problem. The general problem is the dependence of categorization on well-balanced ground truth. Incomplete ground truth may result in overfitting and bad generalization behavior.

In the next two chapters we introduce sophisticated algorithms for categorization that have an excellent balance of performance and model complexity. The next chapter focusses on separators while the majority of those introduced in Chapter 19 are hedgers.

Chapter 18

Risk Minimization Methods

Introduces principles of risk minimization, derives the support vector machine as the optimal implementation of the structural risk principle, lists some of its alternatives, explains the kernel trick and gives examples for kernel functions.

18.1 Risk Minimization Principles

This chapter introduces and discusses a classic model of categorization. *Risk minimization* is one of the central paradigms of machine learning. We endeavor to give a fair overview over the field that explains the fundamental hypothesis, the main practical goals, the state-of-the-art algorithms that implement the goals and, alongside, the concept of feature space transformation by kernels. Risk minimization has been explained many times before in a large number of publications. Our contribution lies in comparing kernel functions to similarity functions. We will see that almost any function of two variables can be a kernel. We investigate the gaps in the set of standard kernels and suggest novel kernels based on psychological insights into the nature of human similarity perception, i.e. methods based on the concepts discussed in the last chapter.

The first section introduces the terms *risk* and *risk minimization*. We investigate the various definitions of risk in machine learning and align them with the fundamental learning problems. In the second section, we introduce the categorization model that is most closely linked to risk minimization: the support vector machine. We explain its origin, optimization criterion and the analytic solution. We will see that the support vector machine establishes a beautiful balance between loss minimization and computational complexity. For opti-

mal performance, however, it requires feature space transformation by kernels. The latter concept is explained in the third section of the chapter. We list the standard kernels, discuss their common foundations and give an outlook on non-standard kernels and future developments. In the last section, we discuss various variants of the support vector machine as well as an approach that is half way between categorization and its evaluation: linear discriminant analysis.

The discussion of this section starts with establishing the *empirical risk minimization principle*. In the second step, we extend it to the *structural risk minimization principle*, which takes computational complexity into account. Eventually, we investigate a few recently introduced alternatives to the state-of-the-art loss functions.

Machine learning by risk minimization is, as a research discipline, heavily indebted to Prof. Vapnik. In [380] he lays down many of his contributions. Below, we follow the general line of his argumentation, only deviating where recent advances have moved the scientific frontier significantly.

The idea of risk minimization is that a classifier should be trained with the goal to make as few as possible *classification errors* on the test set as possible. We assume that such a classifier will perform optimally on unknown data. Hence, risk minimization depends on the availability of a ground truth and implies the usage of classifiers that use a training step. Vapnik defines the *risk functional*, the operationalization of the basic assumption as follows (from [380], p. 18, simplified).

$$r = \int l\big(\mathrm{gt}(f_i), \mathrm{classify}(f_i)\big)\, di \tag{18.1}$$

Here, r is the risk, f_i are the members of the test set, *classify* is the classifier, *gt* retrieves the ground truth value of f_i and l is a *loss function* that defines the penalty for misclassifications. As we can see, the risk is just the sum of the misclassifications of the training set.

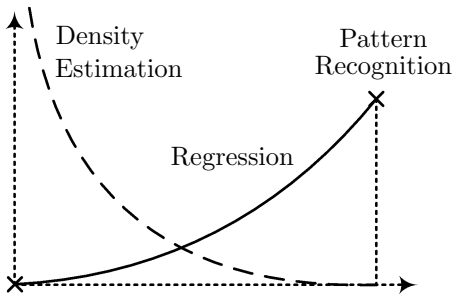


Figure 18.1: Typical Loss Functions.

The essential element in the risk functional is the loss function. Figure 18.1 shows three typical examples and associates them with the fundamental learning problems. Vapnik defines the three depicted loss functions as follows.

$$l_{pr}(x, y) = 1 - \delta(x, y) \tag{18.2}$$

$$l_{reg}(x, y) = m_{L_2}(x, y) \tag{18.3}$$

$$l_{de}(x) = -\log p(x) \tag{18.4}$$

The goal of pattern recognition is perfect recognition of a stimulus. The loss function of pattern recognition (pr) makes use of the Dirac delta function. If the two parameters are equal, this function returns *1* otherwise *0*. Hence, the loss for each misclassification is *1*. The pattern recognition loss function is illustrated by the two x in the figure.

The loss function for regression (reg) is implemented by the Euclidean distance (L_2 norm). The goal of regression is to find a well-balanced classifier for all inputs. Since the sum of squared distances is also used in linear regression it is a natural choice for the loss function. The effect is that large misclassifications cause over-linear losses while small errors are abated. The loss function for regression is shown as a solid curve in Figure 18.1.

Density estimation (de) is a problem fundamentally different from pattern recognition and regression. Here, the goal is to identify a density function – for given descriptions – that is of overall good quality. In the first part of the book we declared that the ideal form is uniform distribution – if the media data allows it. Every other distribution exhibits redundancy. The loss function suggested by Vapnik is based on Fisher's approach which is an early form of the entropy function (see Chapter 22. In this form, every deviation from the uniform distribution is punished. Small values for probability bins cause large losses. These characteristics are expressed by the dashed curve in the figure.

Now, the *empirical risk minimization principle* that represents the practical machine learning problem can be stated in the following forms.

$$r_{pr,reg} = \frac{\sum\limits_{i,j \in L} l_*(f_i, f_j)}{|LxL|} \rightarrow \min \tag{18.5}$$

$$r_{de} = \frac{\sum\limits_{i \in L} l_{de}(f_i)}{|L|} \rightarrow \min \tag{18.6}$$

The empirical risk is summarized over all existing samples. Hence, the ideal integral is replaced by a sum. For each possible categorization the loss function

is computed and summed up. The result is normalized over the number of operations. That is, the empirical risk equals the number/amount of misclassification that a classifier performs on the test set after training.

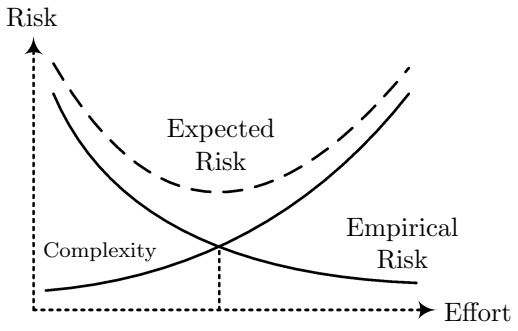


Figure 18.2: Structural Risk Minimization Principle (from [380]).

The empirical risk minimization is naïve in two respects. Firstly, it does not consider the overfitting problem. A classifier that fits perfectly to an awkward ground truth will receive a zero empirical risk even though it will probably fail on real-world data. Secondly, this definition of risk does not consider algorithmic complexity. A complex algorithm with bad performance will be rated more desirable than an algorithm with little worse empirical risk but significantly better performance. Vapnik criticized this ignorance and suggested replacing the empirical risk minimization principle by a dual model. Therefore, he defined the additional goal *confidence* which covers the complexity issue. The confidence in a classifier is low if the algorithm is (over-)simple and high, if it is complex.

Figure 18.2 illustrates the relationship of empirical risk and confidence. For the sake of simplicity we write *complexity* instead of *confidence*. Based on the relationship, Vapnik formulated the *structural risk minimization principle* which states that the ideal classifier should provide a balance of empirical risk and complexity: the *expected risk* (dotted vertical line). It is important to note that the structural risk does not only optimize misclassifications and computational complexity but as well the rigidity of the algorithm (overfitting problem). It eliminates too simple algorithms which would be too rigid as well as too complex algorithms which would be prone to overfitting. The balance of the structural risk optimizes three goals.

In the next section we introduce the support vector machine which is a practical consequence of the structural risk minimization principle. However, before we would like to introduce a few alternatives for the risk and loss functions stated above. Most of these functions have been defined recently. It goes without saying that the concrete definition of the loss function has highest influence

on the practical optimization goal of both the empirical and the structural risk principle. The *minimum risk metric* is defined as follows.

$$r = \sum_{i \in L} \sum_{j \in L} p\big(\mathrm{gt}(f_i)|f_i\big).\big(1 - p(\mathrm{gt}(f_i)|f_j)\big) \tag{18.7}$$

Here, p is the conditional probability for the categorization of a sample f_i in a particular class. The probabilities are built from the actual categorization behavior of the classifier. The resulting risk is similar to the human choice model. It represents the relative chances of misclassification (second term) summed up over all samples.

Two other loss functions are the *minimax* function and the *Neyman-Pearson* function. The first minimizes the maximum of false positives and the rate of the misclassifications. The second minimizes the rate of misclassifications while ensuring that the number of false positives remains below a specified level. These loss functions are relevant for the practical evaluation of classifiers. We will encounter them in Chapter 20.

In summary, the structural risk minimization principle optimizes the number of misclassifications, the rigidity of the algorithm and the complexity of the categorization process. In the next section we introduce the par excellence algorithm for structural risk minimization.

18.2 The Support Vector Machine

The *support vector machine* (SVM) can be seen as the practical consequence of the structural risk minimization principle. In this section, we describe its origin, model and solution. We will see that it is surprisingly similar to a number of long established categorization methods. Still, for many applications the performance of the SVM is unbeaten. We start the section with a short motivation, continue with the goal function and the optimization model (macro process), explain the solution as well as a few tricks on the micro level that are necessary for better performance and, eventually, point out a few applications of the SVM.

The SVM is based on two categorization principles:

- *Linear regression*
- The *perceptron* neural network

Linear regression aims at describing a cloud of data points by a linear function. Therefore, the model is a straight line while the optimization principle is *minimization of distances* from the data points to the model. The goal function is equivalent to the empirical risk minimization principle. The model is very simple and minimizes the complexity of the categorization method. Hence,

linear regression may justly be called a suitable approach for structural risk minimization. The major disadvantages of the method are: It is inflexible, i.e. the model is too rigid for most practical problems, and it is prone to outliers. The optimization goal is usually operationalized by the Euclidean distance which, like the statistical mean, is easily biased by noise, errors of measurement and similar problems.

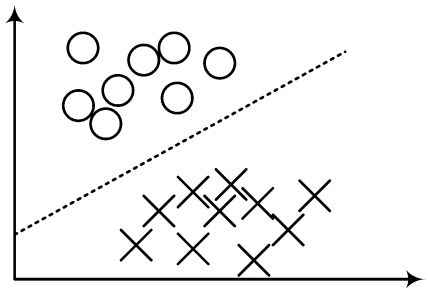


Figure 18.3: Single Neuron Categorization.

The perceptron, in its simplest form, is given in Figure 18.3. On neuron (the dotted line) is able to separate two sets of data points. The equation of this model can be written as $sgn(w_0.f + w_1)$, where f is the description vector of a data point and w_0, w_1 are the parameters of the model. Practically, w_0 is the gradient of the separating line while w_1 is its offset. The perceptron is able to separate a set of points in two groups, hence – though being a separator – it is a suitable model for retrieval problems.

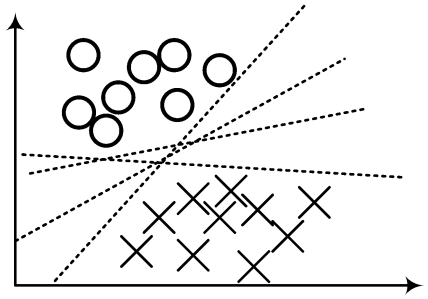


Figure 18.4: Which Separator is the Best?

In the perceptron model, the values for the two parameters depend on the training patterns, the sequence of their presentation and the learning rate. From the point of view of risk minimization, any line that separates the two sets of

data points is a valid solution. Figure 18.4 shows a few examples.

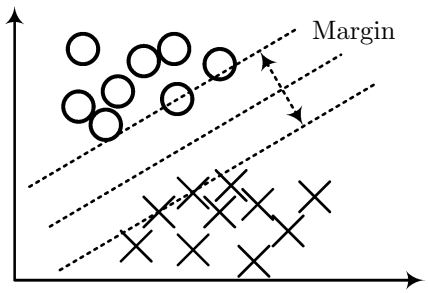


Figure 18.5: Example for Maximum Margin Categorization.

Now, the question is: *Which model is the best?* Vapnik, the author of the SVM, answers this question by introducing the *margin*, defined as the maximum distance of two parallel separators (hyperplanes) that just touch the two sets of points. Figure 18.5 illustrates the idea. Each of the three lines in the figure is a perfect classifier for the problem (zero loss). However, it is easy to argue why the line in the middle is superior to the two others. Since it is further away from the two sets of points it is less likely that a newly added point will violate the model (e.g. an o below the separator). Since the margin is per definition maximal, the center line is farthest away from the data sets and, therefore, at least as good a classifier as any from Figure 18.4 and under consideration of the last sentence in some cases even better.

Before we define the optimization goal of the SVM formally in the next paragraph, we review it beneficial to emphasize that only few data points are required to identify the maximum margin. Precisely, for an n dimensional data space we require $n+1$ points. If we see the borders of the margin as two parallel planes we require n points for determining the position of one plane in the space while we need just one more point for setting the distance of the second plane. The data points that define this system are called *support vectors*. It is an outstanding aspect of structural risk minimization by the SVM that the number of required training vectors is so small. It is the art of SVM optimization, though, to identify these vectors quickly in a large feature space. The next paragraphs show how it is done.

Vapnik states the maximum margin problem formally as:

$$\delta(w_0, w_1) \rightarrow \max \tag{18.8}$$

Here, δ is the function of the margin.[1] The classifier associated with the

[1] For the sake of uniformity and better understanding we do not use the naming conventions

function of the margin implements empirical risk minimization based on the pattern recognition loss function:

$$\text{classify}_{\text{SVM}} = sgn(w_0.f + w_1) \tag{18.9}$$

That is, the parameters of the margin are used to categorize a description f in one of two groups $\{-1, 1\}$. In a similar manner, regression can be implemented with support vectors by using an appropriate loss function (see Section 18.4).

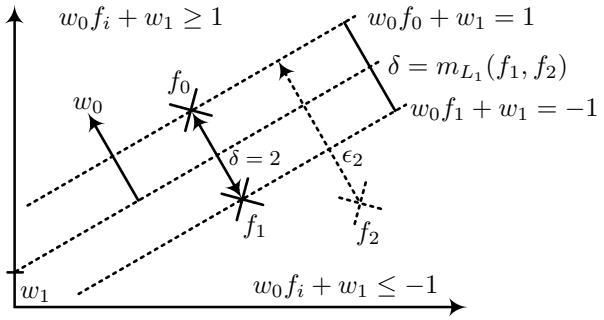


Figure 18.6: Elements of the Support Vector Machine.

How can we express the margin as a function of feature space? The necessary ingredients are gathered in Figure 18.6. First of all, the margin is determined by the vector w_0 perpendicular to the separating hyperplane and by the offset w_1. Then, the margin can be expressed with the help of two description vectors with minimum distance on opposite sides of the borders of the margin, i.e. along w_0. The vectors f_0, f_1 in the figure satisfy this condition. Hence, the margin is the L_1 distance of these points: $\delta = m_{L_1}(f_1, f_2)$. Furthermore, the points f_0, f_1 can be expressed in terms of w_0, w_1 (see upper right of the figure). All points beyond the upper border of the margin can be expressed by the equation $w_0 f_i + w_1 \geq 1$ and analogously for the lower boundary. Our findings so far can be summarized as follows.

$$\delta = |f_1 - f_0| \tag{18.10}$$
$$f_1 = w_0.a + f_0 \tag{18.11}$$
$$1 = w_0 f_0 + w_1 \tag{18.12}$$
$$-1 = w_0 f_1 + w_1 \tag{18.13}$$

introduced by Vapnik to describe the SVM but continue with the notation introduced in the first part of this book.

The second equation states that f_1 can be reached by moving from f_0 in direction w_0 for distance a. Now, we use this set of equations to derive the equation of the margin. After inserting Equation 18.11 into 18.12 and replacing parts of it by Equation 18.13 we reach the result $a = \frac{2}{w_0.w_0}$. Inserting 18.11 into 18.10 yields $\delta = |w_0.a|$. Inserting the first result into the second brings us to the following expression.

$$\delta = \left| \frac{2}{w_0} \right| = \frac{2}{\sqrt{w_0 w_0}} \tag{18.14}$$

The square root of the squared w_0 in the final form is equivalent to the absolute value. Hence, the optimization problem of the SVM can be written as follows.

$$\delta = \frac{2}{\sqrt{w_0 w_0}} \rightarrow \max \tag{18.15}$$

With the conditions:

$$y_i(w_0 f_i + w_1) \geq 1 \tag{18.16}$$

Here, $y_i = \text{gt}(f_i)$ (we have a ground truth!) which allows us to merge the conditions for members of both classes. These conditions need to be satisfied for all vectors f_i of the feature space. For a well separated feature space, of course, the conditions are satisfied for all members if they are satisfied for the support vectors.

But, *what if the two groups of data points are not well separated?* For this problem, Vapnik suggests a standard solution from operations research: the introduction of slack variables ϵ_i. The resulting goal function, known as *soft margin* categorization, is defined as follows.

$$\delta = \frac{2}{\sqrt{w_0 w_0}} - c \sum \epsilon_i \rightarrow \max \tag{18.17}$$

In the equation, c is a constant penalty while ϵ_i holds the absolute value of the distance from the actual position of f_i to the border of the right side of the margin (see Figure 18.6). The soft margin approach is certainly a weak point of the SVM. It works well for classes that are not too closely interwoven. In the next chapter we will discuss approaches that are superior over the SVM for overlapping data. For the sake of simplicity we will omit the penalty term of the goal function in the rest of the discussion.

For solving the SVM optimization problem, we use the Lagrange approach and formulate the *dual optimization problem*. First, we merge the conditions with the goal function by adding Lagrange multipliers a_i. We use the following goal function, which equivalent to the one above.

$$\frac{w_0 w_0}{2} \rightarrow \min \tag{18.18}$$

This trick will help us below to simplify the optimization expression. The Lagrange approach looks as follows ($a_i \geq 0$).

$$L_{w_i,a_i} = \frac{w_0 w_0}{2} - \sum a_i \big(y_i(w_0 f_i + w_1) - 1\big) \tag{18.19}$$

That is, every violation of a constraint causes a penalty. The Lagrange function has to be minimized in w_i and maximized in the Lagrange multipliers. The next step is the elimination of all variables from L except the Lagrange multipliers. Since we are interested in the optimum, we can set the first derivate in directions w_i zero and use the resulting expressions to eliminate these variables in L.

$$\frac{dL}{dw_0} = w_0 - \sum a_i y_i f_i = 0 \Rightarrow w_0 = \sum a_i y_i f_i = 0 \tag{18.20}$$

$$\frac{dL}{dw_1} = \sum a_i y_i = 0 \tag{18.21}$$

Equation 18.20 makes clear why it was a good idea to replace the original goal function. These expressions – that must hold for the optimal solution – can be used to remove w_i from L. After setting $w_0 w_0 = \sum a_i y_i f_i \sum a_j y_j f_j$, some algebra and reordering we arrive at:

$$L_{a_i} = \sum a_i - \frac{\sum_{i,j} a_i a_j f_i f_j y_i y_j}{2} \rightarrow \max \tag{18.22}$$

In this optimization problem, the support vectors are represented by the non-zero Lagrange multipliers. Why is that? In order to be maximal the conditions (second term of L) must not be violated. Every violated condition must be eliminated by a zero multiplier. Hence, what remains are the support vectors. Quadratic programming is used to solve the optimization problem. This part of the solution – the actual training process – is not specific for the SVM.

So far, we have defined the SVM as a very rigid classifier that uses slack variables in order to avoid underfitting. The model is similar to the perceptron and linear regression. The solution, though artful, has the drawback that quadratic programming is required to identify the optimum. This part of the computation is very resource-consuming.

One further trick on the micro level makes the SVM highly effective for the categorization of high-dimensional media data. As we see in Equation 18.22, description vectors appear only in pairs $f_i f_j$ linked by multiplication. The *kernel trick* is to replace this term by some function $k(f_i, f_j)$ that maps the description

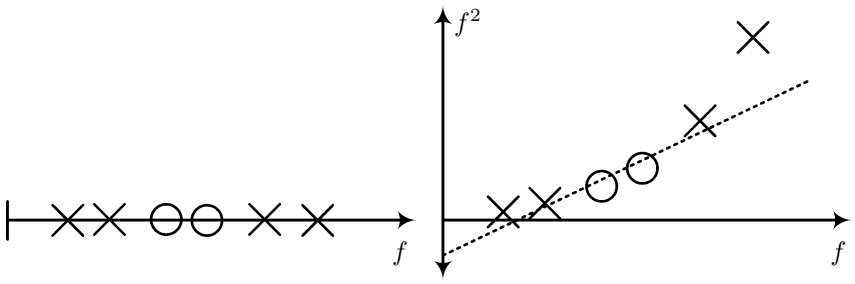


Figure 18.7: Kernel Trick.

vectors to some space of higher dimensionality and measures their similarity there.

Why should we want that? Above we said that the introduction of slack variables for soft margin categorization dilutes the rigid scheme of the SVM. An alternative that allows to separate overlapping data sets by a simple hyperplane would be desirable. The kernel trick provides this alternative. Figure 18.7 explains the principle. The left side of the figure shows a one-dimensional feature space that cannot be separated by a hyperplane at zero loss. The right side shows the same data set with a second dimension that is just the squared first. The right feature space can easily be separated by a hyperplane.

SVM and kernels are closely linked. The model of the classifier is simple, often too simple for complex data sets. Mapping feature space to higher dimensionality introduces white spaces between neighboring points that increases the chance that the space is separable. If it is still not, the slack variables are used. See the next section for a detailed discussion of kernels.

The SVM – as a general-purpose classifier – can be used to separate arbitrary data sets though the performance is not equally good for all problems. The particular strengths of the approach are identifying a reliable solution at good performance due to ignoring the majority of the data points. By nature, the SVM is a separator. The resulting dichotomy is a retrieval solution. For browsing, SVM cascades have to be designed. Since the margin lies in the middle of the space bordered by the support vectors, the classifier does not consider large differences in magnitude of the two classes under consideration. In the last chapter (e.g. Krumhansl density model) we saw that such considerations are important for human similarity perception. Media understanding is the attempt to imitate human perception. Hence, though the output fits naturally it is not advisable to use an SVM for retrieval tasks, because the imbalance of the two classes is not well represented. We recommend using groups of SVM for browsing if cluster analysis shows that clusters of comparable size exist in the training set.

Under such conditions, the SVM performs well.

In summary, the SVM is a fast and stable categorization method with good performance in many situations. Its rigid model makes it a worth implementation of the structural risk minimization principle. Kernel mapping allows to classify even overlapping data sets well. In the next section we investigate the kernel trick and its consequences in detail.

18.3 Kernel Functions

In this section, we review the kernel approach to similarity measurement. We have seen in the last section that kernels can be extremely helpful for the reorganization of a feature space. Like a similarity measure, a kernel organizes a space along the requirements of some object of interest (e.g. a query). Below, we first discuss the fundamental ideas behind the kernel trick. Then, we state the formal requirements of kernels and list the most common solutions for quantitative data. The third part of the section introduces string kernels as a symbolic form that gains increasing attention in text understanding. Eventually, we discuss the usability of similarity functions as kernels, where we make use of the insights gained in the last chapter.

The kernel trick has two components. It is important to note that both components are required to define a kernel function.

1. A *mapping function* creates new dimensions and maps the input vectors to the higher-dimensional space.

2. A *similarity function* measures similarity in the higher-dimensional space.

There is no trick in the second component and the trick in the mapping is very simple. By adding new dimensions but leaving the number of data points constant we necessarily increase the white spaces between the data points. Hence, separation by a model as simple as a hyperplane becomes more likely.

It is not necessary that the mapping function is stated explicitly. Most kernel functions employed today mix mapping and similarity measurement to one function. In the last part of the section we will see that this causes serious limitations for the approach. Almost any function in two variables can be a kernel function. The only formal criterion is *Mercer's theorem*, which states that a kernel function has to be symmetric and positive semi-definite, i.e. $k(x, y) = k(y, x) \geq 0$.

Figure 18.8 plots four major types of kernel functions relative to their similarity measurement functions. These kernels have the following equations.

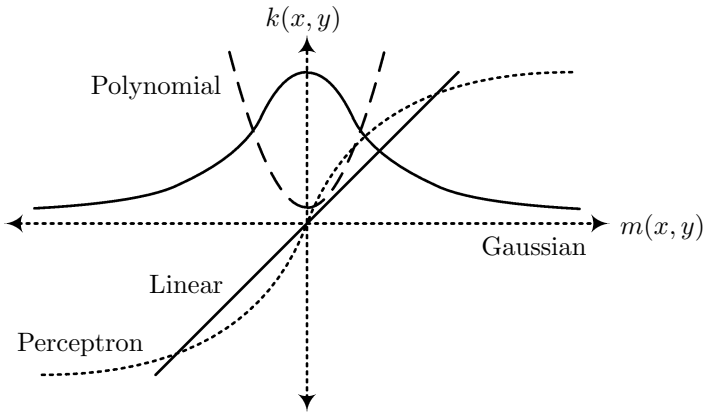


Figure 18.8: Characteristic Kernel Functions.

$$k(x,y) = x'y \qquad \textit{linear} \qquad (18.23)$$

$$(1 + x'y)^a \qquad \textit{polynomial} \qquad (18.24)$$

$$e^{-a(x-y)^2} \qquad \textit{Gaussian} \qquad (18.25)$$

$$\tanh(w_0 x'y + w_1) \qquad \textit{perceptron} \qquad (18.26)$$

The linear kernel has no mapping function. The similarity measurement function is equivalent to the cosine measure and positive convolution. The linear kernel is the standard case in the SVM.

The polynomial function is a real kernel in the sense that mapping is performed. Parameter a determines the number of new dimensions that are created. Similarity measurement is, again, performed by positive convolution.

The Gaussian kernel is interesting because of its similarity to the generalization functions discussed in the last chapter. It does not really provide a mapping into a higher space but weights the similarity measurement according to some Gaussian learning environment. Interestingly, the Gaussian mapping function is typically combined with the L_1 distance measure, i.e. negative convolution. Hence, similarity is here created by generalizing distance. In the last chapter we saw that the Tenenbaum generalization function is the most accurate generalization function today. It would, therefore, make sense to use this function in such a kernel function.

The perceptron kernel imitates a neural firing function for which a Sigmoid curve is used. The parameters of the mapping function define the sensitivity of the neuron. The kernel uses positive convolution for similarity measurement.

These kernels represent four fundamental functions. It has to be noted that the polynomial kernel is the only one that implements the mapping idea fully. The other kernels rather transform feature space to the characteristic topology of a media understanding problem.

Before we continue the general discussion of kernels and give some alternatives to the ones listed above, we would like to introduce two *string kernels* as a particular class of kernels that are used for text understanding today. See [251] for more examples.

The *bags of words* string kernel is defined as follows.

$$k(x,y) = \frac{\sum f_1(i).f_2(i)}{\sqrt{\sum f_1(i)^2 . \sum f_2(i)^2}} \tag{18.27}$$

For two given word histograms f_1, f_2 that hold in element $f(i)$ the relative importance of the ith term, the inner product is computed, i.e. positive convolution is performed. The result is a straightforward similarity measure. This kernel does not contain a mapping function. It is, in fact, a similarity measure.

Another interesting string kernel is the *string subsequence kernel* which implements a form of structural alignment. It is defined as follows.

$$k(x,y) = \sum_{z \in \Sigma^n} \sum_{i \in subseq(z_x)} \sum_{j \in subseq(z_y)} a^{l(i)+l(j)} \tag{18.28}$$

Here, a is a parameter in $]0,1]$. The strings i, j are drawn from all subsequences of terms z from the set Σ^n that consists of all sign-based n-grams of length n. Function $l(i)$ computes the length of string i, i.e. the number characters of i plus intermediate characters that do not belong to the pattern. The term z_x refers to the actual occurrences of string z in document x. Taking $a \leq 1$ to the power of larger than one length values leads to maximal similarity where the lengths are equally short in the two documents x, y. This is guaranteed since we are summing up over all possible combinations. This exhaustive method creates an overall score of the similarity of two documents.

String kernels show that almost any function can be a kernel. We saw that similarity measurement is often performed by positive or negative convolution. However, any other similarity measure could be used as well. If a distance measure is employed, we require a distance to similarity conversion function. Nothing speaks against using Shepard's or Tenenbaum's generalization function. Furthermore, it would be interesting to see the performance of a kernel function that employs normalization by the human choice model on the similarity scores. In a similar fashion, some similarity meta model could be set on top of the similarity measurement process. It would also be thinkable to use a predicate-based similarity measure in combination with some fuzzy interpretation of quantities as a kernel. Such extensions of the similarity measuring part of kernels have

hardly been investigated so far. We are positive that significant performance gains could be achieved by implementing such models.

Concerning the mapping part of the kernel trick we would like to point out that combining existing description elements by a general, data-independent scheme (e.g. using the squared values) is probably only the second best solution to mapping. Instead, the best mapping should be identified by combining promising description elements and evaluating their combined performance *for the given data*. In earlier experiments, the author has found out that systematic selection of description elements that explain the ground truth well (e.g. by canonical correlation analysis, see Chapter 20) and multiplicative combination leads to 'super-dimensions' that can be used for highly effective mapping of feature spaces. In Chapter 20 we will develop this idea further and introduce an evaluation measure for this selection procedure.

The selection of the best kernel for a given problem is an empiric problem. It is advisable to try all standard kernels on a training set and use the one that performs best. With the extensions sketched above the space of possible optimization becomes rather large. Therefore, in a first step only the main options should be evaluated and the details should be set in a fine-tuning iteration.

Kernels are not only used in the support vector machine but have found way into various methods for information filtering and categorization. Generally, a kernel can be used wherever two description vectors are compared. In the next section we will introduce linear discriminant analysis, a method somewhere between categorization and evaluation, where kernels are used. Principal component analysis can be enhanced, if the covariance matrix $\chi = F'F$ is not computed of the elements directly but through a kernel: $\chi = k(F, F)$. If the kernel function is able to handle non-linear data, this type of analysis becomes applicable for such data as well.

In conclusion, every symmetric positive function can be a kernel. Most relevant are mapping functions that add – cleverly defined – dimensions and similarity functions that measure similarity like humans do and as the ground truth represents it. We encourage the reader to combine existing generalization functions with the distance measures in the appendix in order to arrive at new, tailor-made kernels for media understanding.

18.4 Advanced Risk Minimization Methods

In this section we introduce some further risk minimization techniques, most of which are derived from the support vector machine. First, we discuss two derivates/applications of the SVM: the one-class neighbor machine and support vector regression. Then, we describe the structured SVM, an alternative to using cascades of SVM instances for browsing applications. Eventually, we explain

linear discriminant analysis as a classifier. The latter method as a performance measure is discussed in Chapter 20.

The *one class neighbor machine* is a variant of the standard SVM which considers the two classes to be *normal* samples (1) or *abnormal ones* (−1). In practical application, the one class neighbor machine is often combined with domain-specific similarity measures (kernels). For example, in text classification it is common to use the Hamming distance, which is inverse to the city block metric in predicate space, as a kernel function for word similarity.

Support vector regression (SVR) combines the standard model of the SVM with the regression loss function, which results in the following optimization problem.

$$L_{a_i} = \sum a_i - \sum_{i,j} (a_i - a_i^*)(a_j - a_j^*) f_i f_j \rightarrow \max \tag{18.29}$$

We can see two major differences. SVR does not require a ground truth – of course, since we want to approximate the data. The approach is, again, to rely on the support vectors, i.e. those with non-zero Lagrange multipliers. The second difference lies in the multipliers. Here, those referring to Constraint 18.13 are denoted as a_i^* while those referring to Equation 18.12 are denoted as a_i. That is, we aim at a set of support vectors (that define the regression line) that produce a minimal squared error.

The SVR is a very efficient implementation of regression. It can be computed quickly without the need of minimizing the total squared errors. Since the method relies only on few support vectors, it is less prone to outliers than the mean-like linear regression approach. Furthermore, the usage of kernels for the $f_i f_j$ product allows the computation of a regression for non-linearly structured data with a simple computation scheme. Today, SVR is employed in many media understanding applications where the output is used as input for another feature extraction and categorization cycle. One example is the recognition of emotions in audiovisual content that is used for semantic categorization of video clips. Here, SVR provides the emotion cues which are made subject to another categorization process.

The *structured SVM* aims at overcoming the limitations of two-class categorization. Instead of using cascades of support vector machines, the following goal function is suggested.

$$\delta = \frac{2}{\sqrt{w_0 w_0}} - c \sum \max_{c \in C} \Big(m(y_i, c) + w.f(f_i, c) - w.f(f_i, y_i) \Big) \rightarrow \max \tag{18.30}$$

This is the goal function without slack variables. The second term adds a penalty for dissimilarity of the ground truth category y_i to the best-fitting

member of a set of categories C (m being some distance function). It has three components: the similarity of the ground truth category to members of C, a feature score for the distance of the description f_i to the category c (positive influence) and one for the distance to y_i (negative influence). The term *feature score* needs an explanation. This is a feature transformation f applied on the joint vector of description data f_i (or even the underlying sample data o_i) and the category. That is, the semantic value of the category is used for description extraction. Of course, the reasonable definition of function f depends heavily on the application of the structured SVM. It is, in summary, a straightforward extension of the SVM for multi-class categorization.

Before we close the chapter with linear discriminant analysis, we would like to mention the *relevance vector machine*, which is only in name and application similar to the SVM. The basic idea here is, too, categorization in the style of linear regression. The model and training, however, are fundamentally different. The relevance vector machine is based on probabilistic inference. Parameters are estimated from conditional probabilities. The essential training step is parameter learning which is based on an expectation maximization scheme. Hence, the model is more similar to those approaches discussed in the next chapter than to risk minimization, which – as a principle – is not considered in this classifier.

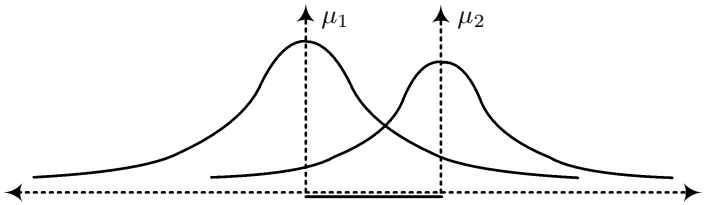


Figure 18.9: Principle of Linear Discriminant Analysis.

Linear discriminant analysis (LDA) is a simple categorization scheme that distinguishes two classes by their means. Figure 18.9 illustrates the principle. Two classes (their descriptions) are assumed to be of Gaussian shape, which allows to describe them roughly by their means. The training process of LDA is the approximation of these distributions. Then, the categorization is based on a simple decision rule.

$$(f_i - \mu_1)\Sigma(f_i - \mu_1) < (f_i - \mu_2)\Sigma(f_i - \mu_2) \rightarrow y_i = 1 \,\text{else}\, y_i = 2 \qquad (18.31)$$

Here, f_i is the object under investigation, y_i is its class, Σ is a covariance matrix of the description elements. Hence, LDA applies the Mahalanobis distance to define a maximum likelihood classifier.

LDA is today hardly used as a classifier, but rather as a performance measure. *Fisher* LDA is a signal-noise ratio based on the LDA idea.

$$\text{SNR} = \frac{m(\mu_1, \mu_2)}{\sum_{i,j} m(\mu_i, f_{ij})} \rightarrow \max \tag{18.32}$$

Here, m is some distance measure and f_{ij} refers to those description vectors f_j that belong according to LDA categorization to the class defined by μ_i. That is, SNR is optimal if the distance between classes is maximal and the distance of class members to the class center is minimal. This goal is highly similar to the one of canonical correlation analysis, which is discussed in Chapter 20.

LDA benefits from the application of kernel functions. If we replace the distances $f_i - \mu_j$ by an appropriate kernel function, LDA becomes able to categorize non-linear data as well.

In conclusion, this chapter introduces a number of separating categorization methods that have in common that they try to minimize the risk of misclassification. Most of these models are very simple and rigid, but make use of kernel functions for the separation of data sets with a sophisticated topology. The common disadvantage of risk-based methods is their dependency on high-quality ground truth. Their advantages are excellent performance due to minimization of the dimensionality problem. Support vector machine and support vector regression are, furthermore, very robust against outliers, since they rely only on those data vectors that separate the classes of the ground truth. Linear discriminant analysis connects the risk-based separators to the hedgers discussed in the next chapter where dynamic methods are used to construct statistical descriptions of semantic categories.

Chapter 19

Optimization Models

Starts with an introduction to fuzzy retrieval methods, explains the self-organizing map, boosting algorithms, mixture models for categorization and density estimation, and closes with a sketch of important global optimization techniques.

19.1 Fuzzy Similarity Measurement

In this chapter, we describe categorization methods that extend the fundamental methods introduced in the first part of the book by a learning algorithm that tries to adapt the categorization model optimally to reality/ground truth. The expert reader will find the list of methods heterogeneous. We start with fuzzy information retrieval (this section), continue with meta-algorithms for learning (next section), introduce mixture models for the representation of real-world distributions of properties (Section 19.3) and close with global optimization algorithms that can be used for categorization. All of the presented methods have in common that they are *meta-models* based on simple categorization schemes. The fuzzy methods presented in this section extend similarity measurement in the vector space model. The self-organizing map introduced in the next section extends the k-means algorithm. Gaussian mixture models merge the idea of expectation maximization with description normalization. Etc. Not all of the presented models are practical categorization algorithms. The core idea of fuzzy retrieval is a form of similarity measurement. While the self-organizing map is a concrete classifier, boosting (also explained in the second section) is a meta-model that can be implemented with any set of (weak) classifiers. Gaussian mixture models may be seen as a form of density estimation as well as a

quantization-based classifier. The global optimization algorithms are fundamental schemes that can be employed for model-based categorization (e.g. dynamic programming as reinforcement learning) as well as on other optimization problems.

The last chapter discussed methods that try to achieve optimal learning by a model as simple as possible. The methods in this chapter aim at building the best possible model and to believe rather in the prognosis of the model than in the given information.

This section targets at the *fuzzy information retrieval* model. Mostly applied in text understanding, it can be used wherever logical expressions (*boolean retrieval*, e.g. in the form of decision trees) are used for categorization. The fuzzy model reduces the rigidity of this form of categorization.

While the standard decision tree assumes all conditions to be AND-connected, the boolean retrieval model allows in the simplest form also OR connections. An OR connection would be equivalent to two unconnected subtrees in a decision tree. The two basic boolean operators are defined as shown on the left side of Figure 19.1. For a given media object, the co-existence of two description elements is defined as the intersection of all occurrences of the stimuli related to the description elements. The OR connection is defined as the set that includes all occurrences of one or both stimuli.

The principle can be understood best from an example. Let o be a text document described by terms f_i. If we want to measure the similarity of o to a query o_q by boolean retrieval (o_q is, for example, 'house AND price OR flat AND rent'), we require a similarity function $m(o, o_q)$ that counts the co-occurrences of the AND-connected terms first and then, sums them up for the OR expression. The higher the resulting score, the more similar o to o_q. Hence, boolean retrieval is equivalent to a forest of decision trees for binary predicates.

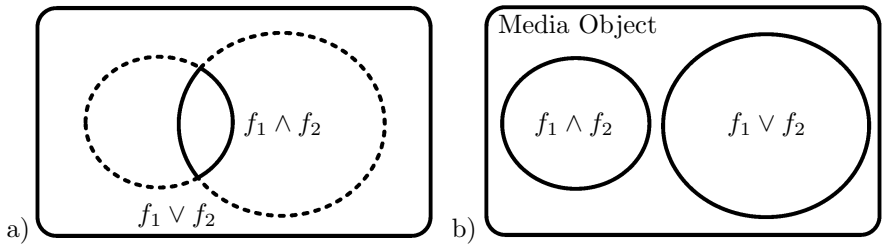


Figure 19.1: Boolean (a) and Fuzzy (b) Similarity Operators.

The *fuzzy retrieval model* extends the scheme of boolean retrieval to description elements that are quantities. This proceeding connects it to the dual process models briefly discussed in Chapter 17. All that is required is the re-definition

of the logical operators. The AND operator is replaced by the *maximum* function while the OR operator is replaced by the *minimum* function. Figure 19.1 illustrates the idea. If we ask for the likelihood of joint appearance of two descriptions f_1, f_2, the fuzzy standard model returns the frequency of appearance of the less frequent stimulus. Instead of the superset, the likelihood of the more frequent stimulus is used for the OR operator.

For the text retrieval example this means that terms are not given as predicates but, for example, as frequencies of the following form.

$$f_i = \frac{\text{number of occurrences of term } i}{\text{number of occurrences of all terms in the document}} \tag{19.1}$$

Then, $m(o, f_1 \wedge f_2) = \min(f_1, f_2)$ and $m(o, f_1 \vee f_2) = \max(f_1, f_2)$. As the figure shows, the definitions of the fuzzy operators may lead to significant deviations in the behavior of the model. If the description elements are predicates, the results of the fuzzy model are similar to the boolean model. However, for quantities the model judges similarity fundamentally different – depending on the relative magnitudes of the logically connected description elements.

The presented model is just one fuzzy retrieval model. Extended forms associate different weights depending on the type of logical connection (e.g. Waller-Kraft model) or depending on the order of the terms (e.g. Paice model). Generally, the approach is similar to the dual process model operators presented in Appendix B.4. As we will see in Chapter 28, these operators try to solve the same problem as the fuzzy operators: transforming a predicate-based model to the domain of quantitative descriptions. The major difference is the importance of the OR operator in fuzzy retrieval, which is irrelevant in dual process models.

The fuzzy retrieval model is mostly used for text retrieval, even though it could be applied on all forms of media data. For example, the model could be used for rule-based video shot segmentation: A wipe could be defined as an OR-connected list of fundamental types (vertical slide in, diamond wipe, etc.). Many more applications are thinkable.

In summary, fuzzy similarity measurement enables the usage of boolean similarity expressions on quantitative data. It is, therefore, better able to represent the polysemy in media objects which goes hand in hand with greater robustness against noise. Similarity measurement happens on the micro-level. In the next section, we move from this level to the macro level and investigate dynamic (learning) extensions of simple categorization methods.

19.2 Learning Meta-Models

We investigate two learning categorization models in this section. The first is the *Self-Organizing Map*, an algorithm based on the k-means classifier. The second

– more general – scheme is *Boosting*, a learning algorithm that can be based on arbitrary classifiers, though it is usually based on simple decision rules (*weak classifiers* or *base classifiers*).

The self-organizing map (SOM) was defined by Kohonen [201] as a two layer feed-forward neural network for unsupervised learning. The neural perspective of the SOM will be discussed in Chapter 29. Here, we focus on the learning algorithm. The fundamental idea of the SOM is that the high-dimensional input data is mapped on a *two-dimensional surface* in a way that similar objects lie in close proximity to each other. The fundamental idea is similar to *multi-dimensional scaling*, as discussed in Chapter 7. As we will see below, the SOM, like multi-dimensional scaling, can easily be extended to n-dimensional output maps.

The model of the SOM is – like the one of k-means – simply a grid of references (also called *codebook vectors*, because they are used to encode/quantize the input vectors). One reference describes one cluster. Since the SOM output map is two-dimensional, it is common to use a rectangular or a hexagonal grid of references. The categorization process for an object o represented by description f is a simple loop over all codebook vectors m_{xy}, where for each reference the distance to f is computed. Typically employed distance measures are the city block norm and Euclidean distance. The reference with minimum distance wins (hence, *winning node*) and the input vector is associated with this cluster.

So far, the SOM is rather a special form of k-means, since it prescribes a particular form of output. What makes the SOM superior over k-means is the learning algorithm. It consists of the following steps.

1. Initialization of the codebook vectors: The vectors may be set to random locations or arranged in the form of a rectangular or hexagonal grid.

2. Repeated learning of all input vectors f_i:

   (a) Identification of the winning node for f_i. In the learning step, the Euclidean distance is usually used for distance measurement.

   (b) Adaptation of the location of the winning node and of its neighbors by the following weighting function.

   $$m_{xy} = m_{xy} - \alpha k(m_{xy} - f_i) \tag{19.2}$$

   Here, α is a learning rate. Typically, the learning process is performed twice, first with higher learning rate (e.g. $\alpha = 5\%$) and then with lower rate (e.g. $\alpha = 3\%$). Function k is a *neighborhood kernel* that penalizes large distances from the reference to the sample. Figure 19.2 shows two examples.

3. Stop the learning process when the map converges.

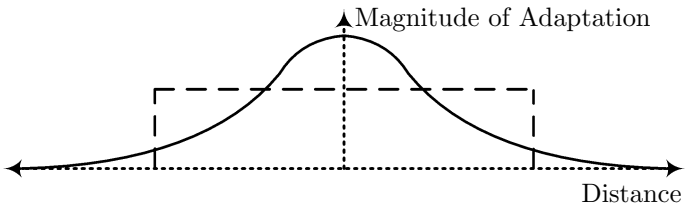


Figure 19.2: Neighborhood Kernels for Self-Organizing Maps.

The strength of the SOM lies in the application of the learning function on the winning node *and* its neighbors as well as in the idea of the neighborhood kernel. The latter function causes that codebook vectors close to the input sample are moved more into this direction than vectors that are far from the sample. Figure 19.3 illustrates the principle. Iterated multiple times over all samples, the SOM gathers more codebook vectors in densely populated areas of feature space than in less densely populated ones. This behavior – also referred to as *neural gas* – is in line with the idea of the Krumhansl density model. The Gaussian neighborhood kernel (solid line in Figure 19.2) implements it perfectly. Alternatively, the *bubble kernel* (dashed line in the figure) implements a k-means classifier with learning references. The learning is the same for all vectors in the neighborhood and zero outside.

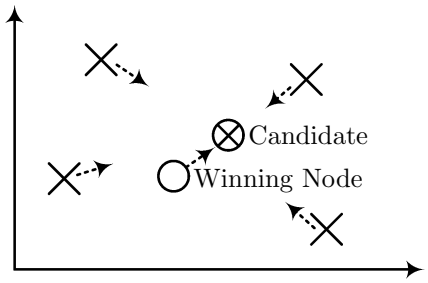


Figure 19.3: Learning of References in the Self-Organizing Map.

When does the learning process converge? When the *quantization error* is minimal. In the context of the SOM, this evaluation measure is defined as follows.

$$q = \sum_i |\bar{m}_i - x_i| \tag{19.3}$$

Here, $\bar{m}$ is the winning node for sample x_i. That is, the quantization error is minimal, if the references are as close to the data as possible (hence, codebook vectors). For the purpose of quantization, the codebook vectors can also be used *instead* of the input data. This form of the SOM is called *linear vector quantization*. A second variant of the SOM is the *tree-structured SOM* that allows to extend each cluster to an entire map. Then, the codebook vector on the higher level represents the mean (however defined) of the map on the subsequent level. Unbalanced tree-structured maps are able to describe the cluster structure of unbalanced feature spaces in great detail.

Above, we mentioned that the two-dimensional approach of the SOM can easily be extended to an arbitrary number of dimensions. The SOM is limited by the grid of codebook vectors and the two-dimensional definition of the neighborhood kernels. Hence, if these two aspects are generalized to n dimensions – which is straightforward, then the algorithm can be used in the same way as multi-dimensional scaling.

Obviously, the SOM is a hedger – on the cluster level where one reference represents one cluster. However, it is not possible to assign a semantic label to one cluster beforehand. All that can be provided is the number of clusters. Instead, the learning algorithm uses as many references as it needs to describe a particular accumulation point in the input data. *Per se*, the SOM is a clustering procedure. The semantic labels can only be assigned after the learning process and will, then, usually span over multiple clusters. Since these groups of clusters are usually not of elliptical shape and sometimes not even joint, the SOM may – on the semantic level – also be classified as an intermediate categorization method.

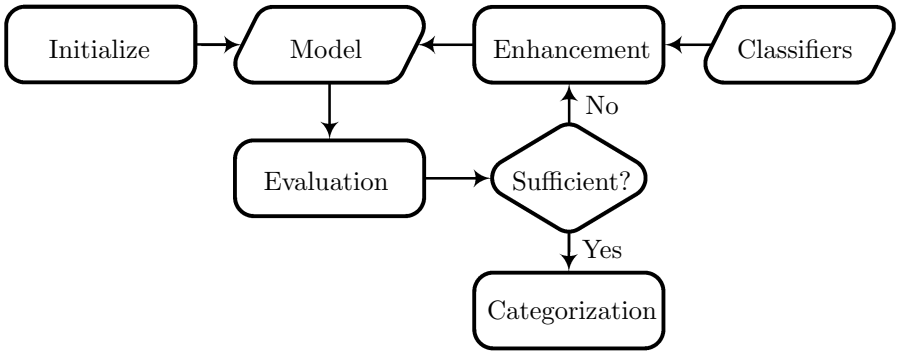


Figure 19.4: Boosting Process.

Boosting is the machine learning meta-algorithm par excellence. Figure 19.4 illustrates the principle. The central element is the *classifier model*, which is a

set of individual classifiers built from a pool of classifiers (rightmost element). In the beginning, the model is initialized empty, with one classifier or with a set of best starting points. Then, the model is iteratively enhanced until a predefined quality criterion is met. This criterion may be based on an evaluation measure (e.g. precision/recall) or a loss function. The enhancement process is typically adding one classifier to the model in one iteration. Of course, the algorithm can also be performed the other way around by initializing the model with all available classifiers and enhancing it by removing badly performing ones. This shows that the basic principle of boosting is actually very similar (if not equivalent) to the one of feature selection. The major difference is the level of operation: contextualization here, summarization there.

Above, we mentioned that any form of classifier can be part of a boosting algorithm. Practically however, simple decision rules of the same form as those used in a decision tree are mostly employed. Such classifiers are applicable as soon as they perform different from chance (success rate of other than 50%). Classifiers performing worse than chance are also applicable, because they can be used in reversed form. This implies that such weak classifiers are primarily used to differentiate the input data in two classes (retrieval application), though the principle can be applied to browsing applications as well. The general boosting algorithm combined with weak classifiers can be employed to build a decision tree for the training data. Since the algorithm includes evaluation, the input data must contain a ground truth. Like the decision tree algorithm introduced in the first part of the book, boosting is generally prone to overfitting.

One concrete implementation of the boosting principle that is comparatively immune against overfitting is *AdaBoost*. This algorithm performs a separation of the input data f into two classes $\{-1, 1\}$ by the following decision rule.

$$\text{classify}_{\text{AdaBoost}} = sgn\Big(\sum w_i.c_i(f)\Big) \tag{19.4}$$

Here, w_i is the weight of the i-th (weak) classifier c_i. The categorization will be optimal, if the following loss function is minimal.

$$\sum e^{-y_j.\text{classify}_{\text{AdaBoost}}(f_j)} \rightarrow \min \tag{19.5}$$

In the loss function, y_j is the ground truth of description f_j. The remarkable application of Shepard's law and multiplicative similarity measurement causes the loss to approach the minimum, if the categorization results match the ground truth perfectly.

In the decision rule, the classifiers are given and the weights are variable. Setting the weights requires a training process. The AdaBoost learning algorithm performs the following steps on an initially empty model.

1. Initialize a set of loss variables $a_j^t = \frac{1}{n}$, where j is the iterator over the n training samples and t is the iterator over the learning cycles. The loss variables express, how well the individual training samples f_j are at time t already represented by the model.

2. Add the classifier c_i from the pool as c_t that minimizes the following expression.

$$l_t = \sum_{j \in \left\{ 1:n \middle| c_i(f_j) \neq y_j \right\}} a_j^t \rightarrow \min \tag{19.6}$$

   The loss l_t describes the contribution of classifier c_t to the model at time/iteration t. If it is zero, the classification is perfect. If $l_t = 1$, all classifications were wrong and $1 - c_t$ is a perfect classifier. If $l_t = \frac{1}{2}$, then the classifier is as arbitrary as a random classifier.

3. For the added classifier c_t, set the weight w_t (equivalent to w_i above) as follows.

$$w_t = \frac{1}{2} \log\Big(\frac{1 - l_t}{l_t}\Big) \tag{19.7}$$

   Please note that $w_t = 0$ if $l_t = \frac{1}{2}$. Better classifiers receive positive weights. Worse classifiers receive negative weights which turns them in the decision rule to positive classifiers.

4. Evaluate the classifier by Equation 19.5. If this expression is below some threshold ϵ, stop the boosting process, otherwise continue with the next step.

5. Update the set of loss variables for the next iteration by the following function.

$$a_j^{t+1} = a_j^t . e^{-w_t c_t(f_j) y_j} \tag{19.8}$$

   Hence, the loss decreases (weighted by w_t) for correct classifications (exponential term smaller than one), and increases otherwise (exponential term greater than one).

6. Eventually, return to Step 2.

AdaBoost is a very elegant algorithm that makes use of the two class labels $\{-1, 1\}$. The additional usage of a generalization function creates human-like similarity judgment. Naturally, the algorithm *separates* feature space. It creates a tailor-made decision tree – if the weights are seen as part of the individual classifiers – that fits as close as possible to the ground truth.

Both introduced meta-algorithms rely on the provided ground truth. If it is incomplete or unbalanced, the result may bo overfitting and inferior performance on real-world data. Furthermore, both approaches are sensitive to noise and outliers. The SOM deviates towards outliers, while AdaBoost creates an inadequate representation. However, the iterative learning procedures implemented in the two algorithms help to rise the semantic level of the categorization results. This semantically higher level is represented by the classification model (references, weights). In the next section, we introduce an approach to represent semantics in complex density functions.

19.3 Advanced Densities: Mixture Models

The estimation of density functions is a fundamental problem of machine learning and media understanding. For example, the application of Bayesian methods requires the transformation of the given training data in a set of confusion matrices (conditional probability distributions). In the first part of the book, we introduced two fundamental techniques for the estimation of density functions: Gibbs sampling and expectation maximization. We did not make any assumptions about the shape of the probability distributions. Please note that the statement that description elements should be uniformly distributed has nothing to do with the probability distributions of joint events (e.g. the likelihood of co-occurrence of a particular query and a description).

In this section, we go one step further by structuring the density functions that should be estimated. We introduce the mixture model concept and use it for direct categorization as well as indirect application in probabilistic inference algorithms. We will see that mixtures fit to recent psychological results about human cognition.

For our purpose, a *mixture model* can be defined as a linearly weighted combination of fundamental probability distributions. Formally:

$$Q(x) = \sum w_i P_i(x, a_i) \tag{19.9}$$

Here, P_i is a probability function of random event x with parameter set a_i. If all probabilities are of the same type (e.g. Gaussian), we speak of a *parametric family* (e.g. means and standard deviations). Obviously, a mixture Q of probability functions with weights $w_i \geq 0$ is convex. By default, we also have the condition $\sum w_i = 1$.

Mixtures can be employed in the same way as other distributions. Their main advantage lies in the form of construction. A mixture built from smooth density functions can very well be employed for the description of the convex hull of a data set that gathers around an accumulation point. Such accumulation points are – as we have seen – typical for feature spaces derived from media data sets. Psychological findings support this view. Therefore, the authors of [182] base their concept of *stimulus norms* – a form of prototypical concept representation (see Chapter 17) – on mixtures of Gaussian probability distributions. The norms are constructed from experience (samples) and used as references in the human cognitive categorization process.

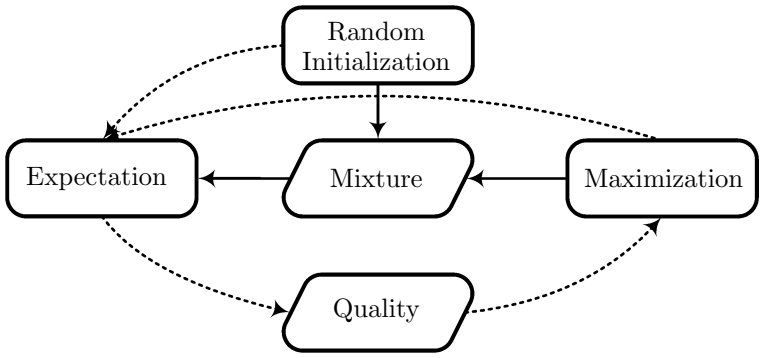


Figure 19.5: Building Process for Mixture Models.

What is good for humans should also be good for machines. The only problem in the application of mixture models for categorization is the estimation of the unknown parameters w_i, a_i. For the sake of simplicity, we assume the type of probability distribution as given (for example, a parametric family of Gaussians). The standard solution to the parameter estimation problem is the *expectation maximization algorithm* as depicted in Figure 19.5. The central mixture model is first initialized randomly, then iteratively used to estimate the model quality (e.g. in the form of categorization performance) and to improve the model based on the results.

In detail, the following steps are performed after the random initialization.

1. Expectation: Estimate the parameters from the training data. For this purpose we define a set of *membership values* y in the following way.

$$y_{ij} = \frac{w_i P_i(x_j, a_i)}{Q(x_j)} \tag{19.10}$$

The membership value y_{ij} can be interpreted as the relative contribution of the i-th probability density to explain sample x_j.

2. Maximization: Refine the parameters based on the membership values. In particular, the weights can be set as follows.

$$w_i = \frac{\sum_{j=1}^{n} y_{ij}}{n} \tag{19.11}$$

That is, the new weight w_i of probability density p_i is the average membership value over all n training samples. The refinement of the density parameters depends on the form of distribution. For Gaussian functions the following rules can be employed.

$$\mu_i = \frac{\sum_j y_{ij} x_j}{\sum_j y_{ij}} \tag{19.12}$$

$$\sigma_i = \sqrt{\frac{\sum_j y_{ij} (x_j - \mu_i)'(x_j - \mu_i)}{\sum_j y_{ij}}} \tag{19.13}$$

Hence, the mean of the i-th density function is just the expected value of the membership values. The standard deviation measures over the squared distances to the mean.

3. Loop: Return to the first step until the categorization quality is above a pre-set threshold.

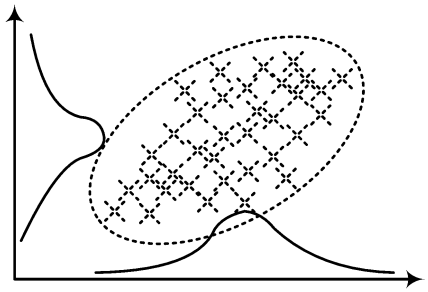


Figure 19.6: Example of a Simple Mixture Application.

Mixture models can, as stated, be used to represent clouds of data. Figure 19.6 shows an example. The resulting description can be used in one of two ways.

- Directly as a classifier

- As the model of a probabilistic classifier

An example for the first usage is the *Gaussian Mixture Model* classifier (GMM). The GMM uses the maximum likelihood principle for categorization. In the training step, each class i is represented by one mixture Q_i. In the categorization step, the input object's description f is weighted by each mixture and the one with maximum likelihood is chosen. Hence, the class label is derived as:

$$\text{classify}_{\text{GMM}} = \arg\max_i q_i(f) \tag{19.14}$$

The only information that needs to be provided is the number of classes n. Before we discuss the advantages and disadvantages of the GMM approach we would like to point out that the second usage is, typically, the computation of the confusion matrices required for a hidden Markov model by expectation maximization of Gaussian mixtures. There, the probabilistic inference is performed as described in the first part of the book.

The major advantage of the GMM is that it fits naturally with the structure of feature spaces derived from media objects. The analogy to human perception and cognition supports this argument. On the other hand, GMM are prone to entering local optima. In the next section we will discuss several algorithms that are able to escape such suboptimal situations. The GMM is not. Furthermore, the expectation maximization algorithm is not deterministic. For suitable data, it will show dynamic – even chaotic – behavior and oscillate between suboptimal solutions for parameters and weights. In Chapter 26 we will discuss the properties of such dynamic systems. The practical consequence is that the GMM algorithm will not always terminate – depending on the random initialization and the input data.

Another problem of random initialization is that the training process may result in two or more mixtures (each one representing a different semantic category) that are very similar to each other. In this case, the training process has to be repeated for different starting points. Since GMM training is time-consuming, the usage of this classifier has a negative effect on the performance of the media understanding process. However, the actual maximum likelihood categorization can be performed very quickly and mixtures are capable to represent the polysemy in the input data well. For these reasons, GMM is a popular categorization method for Gaussian-shaped feature spaces and a popular density estimator for Bayesian methods.

19.4 From Local to Global Optimization

In the last section we move from concrete machine learning techniques to a general problem of categorization methods: escaping from a local optimum towards the global one. This question is relevant, because – as we have seen – the learning algorithms do not solve a strict optimization problem. Instead, the solution depends on the input data and parameters (samples, references, etc.). This statement is equally true for the self-organizing map, certain boosting algorithms and Gaussian mixture models.

Below, we discuss three fundamental *escape techniques*:

- Simulated annealing
- Genetic algorithms
- Dynamic programming

All three *global optimization approaches* stem from operations research, i.e. they are not machine learning-specific and, hence, no categorization techniques. However, many categorization methods are based on these techniques, which is why we consider it beneficial to know their fundamental principles. The discussion in the third part will partially be based on the knowledge presented in this section. Of course, we do not intend to give a full explanation of the algorithms. Rather, we review them from the media understanding perspective.

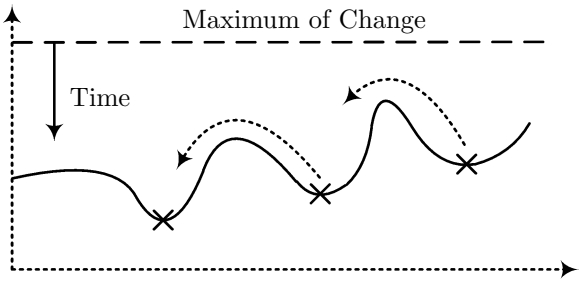


Figure 19.7: Escape from Local Optima.

Figure 19.7 illustrates the escape problem. The central curve represents the goal function of an optimization problem. If we assume a minimization problem, we have two local minima (middle, right) and one global minimum (left). A *brute force search algorithm* that searches for the global optimum from the right (e.g. *hill climbing*) may terminate in the location marked by the rightmost x. Only after considerable time would it reach the global minimum.

The first escape principle that we would like to discuss is *simulated annealing*. The fundamental idea is expressed by the dashed line in Figure 19.7. Whatever optimization algorithm is used, the search is limited over time in a way that the constraints of the search are tightened with increasing time. For example, for the brute force search algorithm the dashed line of maximal change is moved towards zero over time. Depending on the speed, the algorithm may be able to jump over the rightmost hill, but probably not over the one in the center. In consequence, the central minimum would be regarded as the global minimum.

What is the use of simulated annealing? The example shows that it leads to a suboptimal result. Why should we want that? Simulated annealing *limits the search process*, which is a very important property in optimization. Optimization algorithms are distinguished by their computational complexity. Being able to guarantee that they terminate in acceptable time is valuable. Simulated annealing provides this ability in a somehow natural fashion. Over time, the initially *hot* search process freezes in at a – hopefully, good – optimum. The quality of the solution, though, does not depend on the annealing but on the search algorithm. Simulated annealing processes are implemented in various forms. For example, in Chapter 29 we will encounter the Boltzmann machine that uses a cooling scheme for the limitation of the classifier learning process. In the same way, simulated annealing could be used in Gaussian mixture models to avoid chaotic oscillation in the training step. In fact, the reduction of the learning rate in self-organizing maps is a coarse form of annealing.

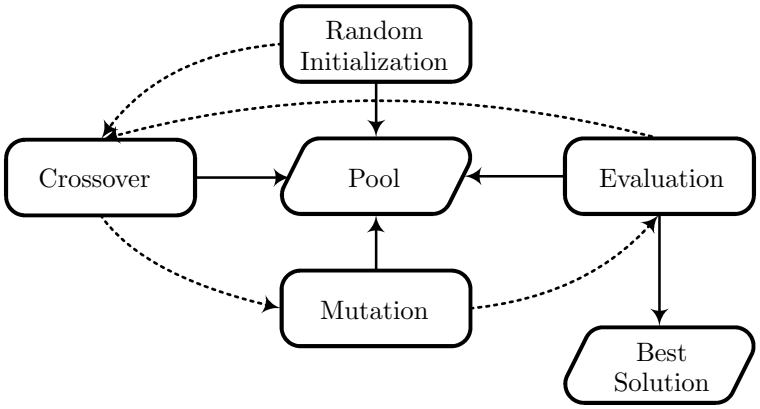


Figure 19.8: Process of the Genetic Algorithm.

The *genetic algorithm* (GA) is one search algorithm that could be combined with simulated annealing, though the author is not aware that this would have been tried so far. The search process is visualized in Figure 19.8. The central

element is a pool of gene strings. One gene encodes one solution for the optimization problem. In the classic form, genes are binary strings. Extended forms allow sequences of symbols.

After random initialization, the GA tries to enhance the gene pool by iterative application of crossover, mutation and selection operations. In a crossover, two gene strings are halved and merged. Mutation means that n per cent of the symbols in one string are mutated randomly. Selection is based on evaluation (typically, by the goal function). The n best gene strings are kept, the rest is thrown away. Repeating this scheme leads over time to improvement in the gene pool, though it is not necessarily the case that the last iteration contains the best solution. Through crossover and mutation, the genome can degenerate which serves as an escape algorithm. The major advantages of the GA are the parallel investigation of search space and the possibility of escaping from local optima through mutation.

The GA can be employed to solve arbitrary optimization problems including categorization. Like a Bayesian network, the GA shifts the complexity from solving a problem to stating it. If the genome is binary, the two operators are well defined. All that is required then, is the definition of the structure of the gene strings and of the optimization function. The latter function is equivalent to the evaluation measure in media understanding. Hence, defining the gene string remains as the only problem. It is an interesting idea that – due to the generality of the approach – almost any categorization scheme can be expressed in a GA genome. Furthermore, it is thinkable to combine the GA with annealing, for example, by cooling down the rate of mutations over time/iterations. This proceeding would increase the probability that the final gene pool includes the achieved optimum.

We would like to close this section with *dynamic programming*, a very general optimization model that is characterized by *divide and conquer analysis*, *recursive search* and *reinforcement learning*. The latter principle was already mentioned in Chapter 17. Recursive search is, for example, used in the inference algorithms of Markov processes, but as well in dynamic time warping and gene sequence alignment. Divide and conquer is a general principle of information analysis closely linked to the *top down strategy*.

In the context of media understanding, Figure 19.9 brings the three principles of dynamic programming together. The problem is, for example, a categorization process. In the first step, the global problem is split into *subproblems*. The division may, for example, be performed along the data (e.g. one solution per group). In the conquer step, the subproblem is solved. The resulting *subsolution* is made subject to evaluation which leads to a *reward*. This is the reinforcement step of the approach. In reinforcement learning, we do not work with ground truth but with penalizing losses based on rewards for correct categorizations. Hence, the optimization goal of the categorization process must be maximization

of rewards. Formally:

$$\sum_t \alpha^t r_t \rightarrow \max \tag{19.15}$$

Here, r_t is the reward at time t (e.g. one application of the classifier) and α is a down-scaling factor for future rewards, which are – for machines as for humans – the more interesting the earlier they are brought in. Reinforcement learning is as independent a principle as the divide and conquer approach and recursive search. It can be implemented in combination with dynamic programming but as well by a brute force algorithm that maximizes the total reward by trying all possible solutions. A practically more relevant form is the description and prediction of desirable rewards by conditional density functions of the form $P(r|a)$, where r is the reward for action a, for example, a particular categorization operation. Following this path, we once again end up in the density estimation problem.

Returning to dynamic programming, eventually, the individually solved subproblems are merged to a global solution. This step is a form of boosting. The other way around, if the (weak) classifiers are interpreted as conquer steps, the AdaBoost algorithm may be seen as a dynamic programming algorithm. The entire algorithm will usually be implemented recursively, i.e. as nested dynamic programming where the optimal solution is derived from rewards for atomic subproblems. As we have seen in earlier chapters, recursive algorithms often lead to superior computational performance, because they manage to alter a fraction of the processing requirements to memory requirements.

In conclusion of this section, the genetic algorithm and dynamic programming are two principal optimization approaches that can be used for categorization in media understanding. In combination with simulated annealing, they can be used to implement almost arbitrary categorization schemes. Generally, the algorithms have a positive effect on the dimensionality problem of media understanding since they search as efficiently for high quality optima as possible. Their common drawback is a tendency to emphasize the low-level descriptions in media understanding, because they follow the topology of feature space. This behavior does not help to close the semantic gap.

This chapter and the last should provide a fair toolbox of state-of-the-art categorization methods for media understanding. SVM and GMM, for example, are used in numerous single-media and multi-media analysis applications today. However, their training requires more sophisticated performance evaluation techniques than those discussed in the first part. Such methods are introduced in the final chapter of the second part.

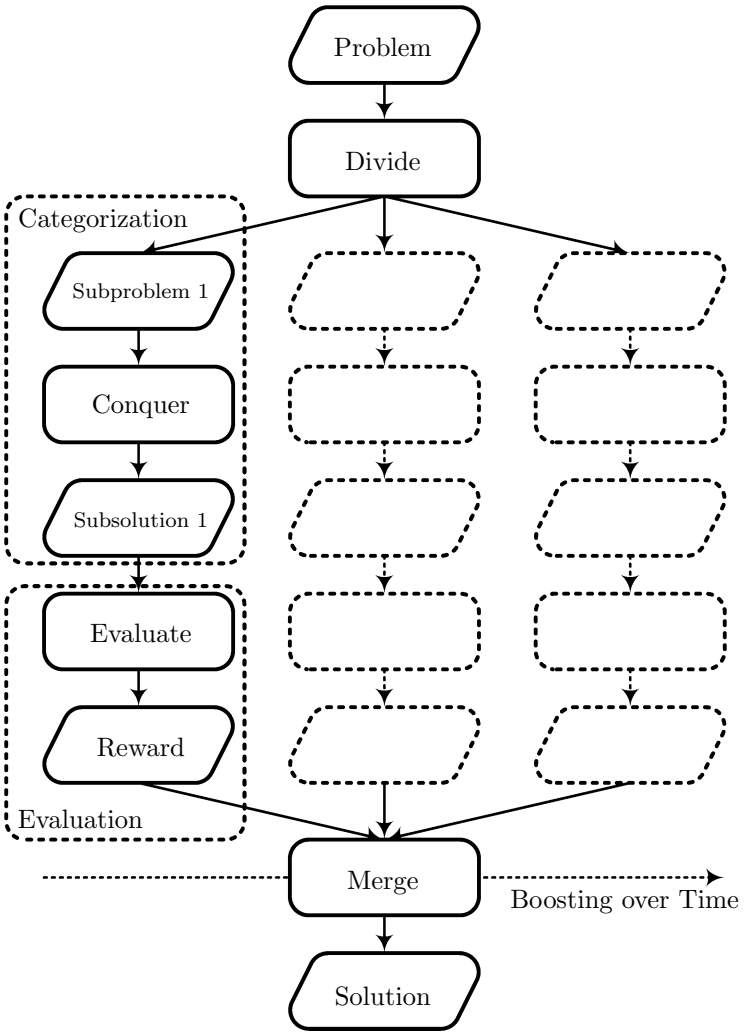


Figure 19.9: Dynamic Programming Process with Reinforcement Learning.

Chapter 20

Advanced Evaluation

Introduces schemes for systematic classifier testing and evaluation, discusses information-theoretic measures for evaluation and suggests measures and processes for evaluation-based description refinement.

20.1 Cross Validation

In this last chapter of the second part we turn our attention to the evaluation problem. Feature transformations summarize media content and relate it to templates. Information filtering improves the data quality of the media descriptions. Categorization reduces descriptions to concepts. Now, the evaluation step aims at the measurement of the practical quality of the media understanding system. In the first part, we already laid the basis and introduced the fundamental components of the evaluation process as well as the most relevant measures.

In this chapter, we generalize the methods introduced in the first part. The first section focusses on the macro process of evaluation. We replace the arbitrary separation of world information into training data and test data by a systematic process of validation. The second section generalizes the measurement process. The ground truth-based measures are positioned in an evaluation framework that can be used to assess arbitrary categorization methods. Section 20.3 abstracts from concrete measures for media understanding quality to general measures for data/information quality. In the last section of the chapter, we return to the practical side of evaluation and introduce a number of systematic measures for the overall quality of the media description process. These measures establish the link back to the first chapter of this part, in which we reasoned over the

properties of good feature transformations.

This section is dedicated to the evaluation process. However, before we go in medias res we would like to emphasize that the evaluation process is independent of the actual measurement goal. The last paragraph had the hidden message that there are two different evaluation problems.

- Performance with respect to world information
- Information-theoretic and statistical data quality

The same process can be employed to evaluate both domains. The first problem is out of question, since media understanding is a practical domain and its acceptance stands and falls with the computer's performance in comparison to the human competitor. Performance measurement based on world information investigates this aspect.

But why should it be interesting to measure the data quality, in particular, of the descriptions? What does it matter if the entropy is bad as long as recall and precision are good? The practical answer is that the two evaluation domains are related. It is very unlikely that a media understanding system produces high F_1 scores for recall and precision if the variance in the descriptions is inferior. Furthermore, often it is not possible to provide satisfactory world information for evaluation. We mentioned several times that providing a well-balanced, complete ground truth is a complex, time- and resource-consuming tasks. The more general the problem domain, the harder the tasks becomes.

Eventually, empirical results are not always trustworthy. Empirical statistics rely on the *law of great numbers*. As we will see in Chapter 23, humans tend to derive from the law of great numbers a *law of small numbers*, i.e. the statistical laws should also be more or less valid for small samples – which is nonsense. Hence, if the ground truth is too limited (though, how much is *too limited*?), then empirical results will not truly be empirical. In this sense, Nigel Barley relates an interesting story in his book *The Innocent Anthropologist* [15]. When he finds out that the rainmakers of the Dowayo tribe use found marbles for the rainmaking ritual, he presents a bought marble to one rainmaker and asks him whether he could also use this one for rainmaking. The rainmaker answers: "How should I know? I have not tried it yet." Empirical results on a too small base of samples are questionable.

On the other hand, the evaluation of data quality is an option for evaluation that is always available and that indicates how good the performance might be in the best case. Moreover, it indicates if the performance in terms of processing power and memory usage is optimal – two factors that are highly relevant in the resource-intensive domain of media understanding.

This stated, Figure 20.1 illustrates the *Cross Validation* process, the general evaluation process based on ground truth information. It is an iterative process

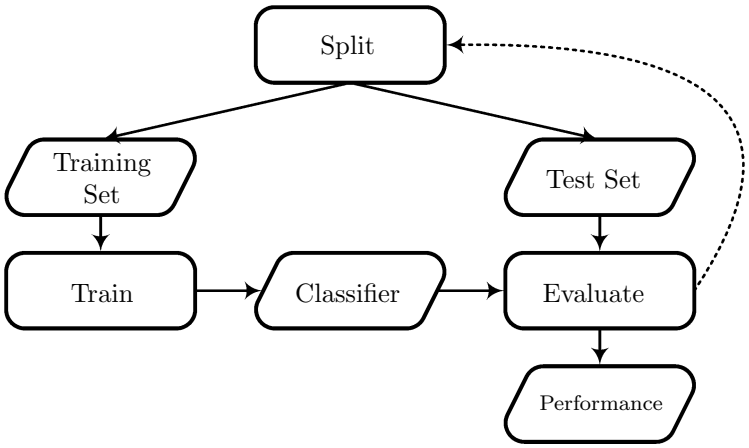


Figure 20.1: Cross Validation Process.

in which the available world information is repeatedly split into a training set and a test set. The first set is employed to train the classifier under consideration. The second set is employed to evaluate the classifier. The resulting performance values are aggregated (e.g. averaged) over the iterations of the validation process.

The essential property of cross validation is that each practically relevant group/concept/class/etc. is evaluated *at least once*. Hence, the resulting performance value should include all practically relevant aspects of the classifier. It goes without saying that the statistical averaging of the individual performance values will only be appropriate if the variance as a form of belief – is included.

Normally, ground truth categories will not be mixed in training set and test set. That is, one group (semantic concept) will either be part of the training set or of the test set. Practically, cross validation with mixing of semantic concepts may be seen as a generalization of the standard approach. In a strict sense, however, cross validation is an approach that measures the ability to predict some semantic concepts from learnt knowledge representations about other semantic concepts (hence the name).

One particular form of cross validation that is practically highly relevant is *Leave One Out Cross Validation* (LOOCV). In LOOCV, the test set of each iteration of the validation process consists of *all members of exactly one class*. The world information about all other classes is used to train the classifier and the remaining group is used to evaluate it. In consequence, LOOCV evaluates the *generalization power* of the classifier, i.e. the ability of the classifier to estimate group membership in the unknown group from the known groups. LOOCV is a widespread method in science, because generalization is one of the most

important abilities of a good classifier. Moreover, the limitation of the test set to one class allows to conclude on shortcomings of the classifier for particular semantic groups. Eventually, the training set will generally be larger than the test set, which provides a bottom line for the learning potential of the classifier. If LOOCV values are bad, they will probably be bad for most other evaluation methods as well.

In conclusion, cross validation is the standard approach for comprehensive evaluation of classifiers. It requires the existence of some form of world information, most typically ground truth labels. Cross validation may be combined with any form of performance measurement. In the next section, we introduce a general graph-based scheme that fits validation naturally, because it does not require the statistical aggregation of performance values.

20.2 Receiver Operating Characteristic Curves

After describing a systematic evaluation process in the last section, we use this one to introduce a systematic measurement scheme. The *Receiver Operating Characteristic* (ROC) curve can be used to visualize the evaluation results gained for parameterizable classifiers as well as simple static categorization methods. First, we discuss the history of the approach. Then, we introduce the contingency table of evaluation as the foundation of ROC analysis. From the table we derive advanced measures and the ROC curve itself. Eventually, we embed the ROC curve in the cross validation process.

ROC analysis is a child of operations research. During the second world war, it was developed as a tool for *signal detection*, i.e. the differentiation of signal and noise in a categorization process. Originally, the approach was developed for *binary classification*. That is the classification in two groups, for example, $\{-1, 1\}$ as performed by the support vector machine. However, the approach can easily be generalized to classifiers that distinguish more than two classes. In fact, this is merely an interpretation problem. We think that it is more precise to characterize the ROC approach as an evaluation scheme for *retrieval applications* rather than for *binary classification*.

The entire approach is based on world information (e.g. ground truth). After training, the classifier under investigation is evaluated based on the number of correct and false classifications it produces. These numbers are normalized by the number of correct and false samples in the test set and, eventually, visualized in a two-dimensional graph.

Table 20.1 is the *contingency table* of ground truth-based evaluation. It summarizes the measures required for ROC analysis systematically. In retrieval, we distinguish *relevant* and *irrelevant* items (the above-mentioned generalization follows at the end of the section). This view is laid down in the ground truth.

True Positives (Hits)	True Negatives (Rejection)	*Correct*
False Negatives (Misses)	False Positives (False Alarms)	*False*
Relevant Items	*Irrelevant Items*	*Sum*
Ground Truth		

Categorization

Table 20.1: Possible Results of Categorization.

From the perspective of the categorization process, we distinguish correctly classified items and falsely classified items. Crossing out these two views results in four measures for the performance of a categorization process on a given ground truth. The *true positives* (TP) and the *true negatives* (TN) define the success of categorization while the false positives (FP, *error of first type*) and the false negatives (FN, *error of second type*) measure its failures.

From the measures in the contingency table, the following relevant measures can be derived.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{20.2}$$

$$\text{Fallout} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{20.3}$$

$$F_1 Score = \frac{\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{20.4}$$

In ROC theory, the recall is also referred to as the *true positives rate*, because the sum of true positives and false negatives gives the number of relevant (positive) items. Fallout is referred to as *false positives rate*, because true negatives and false positives sum up to the irrelevant items.

ROC analysis is a visual method. It makes use of recall and fallout to define a two-dimensional graph. Figure 20.2 shows an example. Both dimensions are per se normalized to the interval $[0, 1]$. A classifier that is positioned along the diagonal dashed line is a random classifier, because it produces an equal number of true positives as of false positives. Hence, every classifier that moves away from the diagonal is usaful. The area above the diagonal contains the positive classifiers – where the upper left corner of the diagram is the optimum.

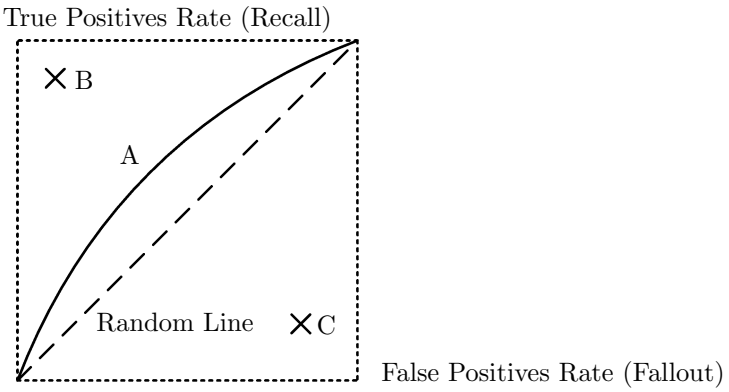


Figure 20.2: Receiver Operating Characteristic Example.

Classifier B, for example approaches the optimum. The area below the diagonal contains negative classifiers that can, for example, be used in the way of the AdaBoost algorithm by taking the inverse of the classification output as the class label. In this sense, the performance of classifier C is comparable to the one of B. Eventually, the curve of classifier A shows the behavior of this classifier for varying parameterization. The assumed case is a decision stump of the following form.

$$f_i < t_i \rightarrow 1 \,\text{else}\, 0 \tag{20.5}$$

Curve A develops by varying threshold t_i for description element f_i. Of course, the approach can easily be generalized by assuming t_i to be a container of parameters relevant for the categorization process. Then, the ROC line would represent all results of a systematic search through the parameter space.

ROC analysis has one obvious advantage. Since it provides a systematic visualization of evaluation measures, the user becomes able to understand any categorization process as soon as he has understood one. ROC analysis shifts the evaluation problem from measurement to visual interpretation. A typical application of ROC curves is the comparative visualization of categorization methods (e.g. evaluated by Weka). The one method and parametrization that comes closest to the upper left or lower right corner can be chosen as the best. Of course, this approach is only valid, if the ground truth is – once again – representative for the real-world problem.

The embedding of ROC analysis in cross validation is straightforward. The results of each iteration of the cross validation process can be visualized in the ROC curve. After some time, a picture of the characteristic performance of the classifier under investigation will emerge. Cross validation and ROC analysis

are two systematic evaluation methods that fit together naturally.

The generalization of the ROC approach from retrieval problems (two classes: positives/negatives) to browsing problems (n classes, each one with potentially correct/false evaluations) requires the redefinition of the dimensions of the ROC graph. We suggest using the weighted average of recall and fallout values over all classes (expected value). For the recall formally:

$$\mathrm{TPR} = \frac{\sum \mathrm{TP}_i}{\mathrm{TP} + \mathrm{FN}} \tag{20.6}$$

Here, TP_i is the number of true positives for the i-th class. TP and FN stand for the total numbers of true positives and false negatives. Since the sum of individual true positives equals the total number of true positives, the definition does actually not change in the browsing case. The only difference is that the performance per group is hidden in the ROC graph.

In conclusion, the value of ROC analysis lies in the uniformity of the approach. Understanding the principle means understanding all applications. In combination with cross validation, ROC analysis is an expressive tool for ground truth-based evaluation. In the next section, we discuss a systematic evaluation approach that can do without world information.

20.3 Information-Theoretic Measures

We criticized it several times in the first two sections of this chapter, but *a comparison of media understanding solutions to human judgment* is what we eventually want to achieve. That given, it would also be desirable to estimate the quality of the data that is being produced in the media understanding process. The straightforward way to do that is by statistical analysis, as we proposed in the information filtering chapters of the first and second part of the book. There is, however, a second approach, that, though related to descriptive statistics, has an independent existence in the world of computer science: *information theory*. As we will see in this section, entropy in information theory and statistical moments are two instances of the group of *interestingness measures*.

In this section, we deal with interestingness measures, in particular, information entropy as defined by Shannon. We start by outlining the original application scenario. Then, we state and discuss the most important measures and relate them to their production processes. Eventually, we compare the information-theoretic measures to already introduced measures (signal-noise ratio, for example) and position them under the umbrella of interestingness measures.

Figure 20.3 illustrates the communication process that was the foundation of Shannon's reasoning about information-theoretic measurement in [333]. A

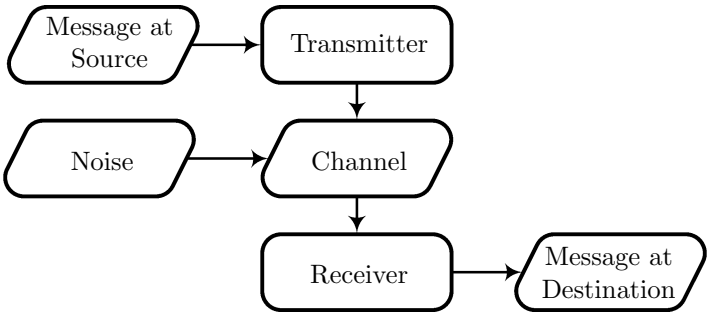


Figure 20.3: Fundamental Model of Information Theory.

message needs to be transmitted from a source to a destination. The transmitter controls the sending process. The receiver grabs the result. Noise pollutes the communication channel. The entire process of communication and denoising can be seen as a *transducer*, i.e. a dynamic system with the following signature.

$$y_t = f(x_t, \alpha_t) \tag{20.7}$$
$$\alpha_{t+1} = g(x_t, \alpha_t) \tag{20.8}$$

The output y_t of the transducer f at time t depends on the input x_t as well as on the parametrization α_t. The update function g of the latter depends on the input and the state. This model of a transducer is applicable to a number of dynamic systems, including the Kalman filter that will be introduced in Chapter 26 as well as the categorization methods discussed so far.

Shannon modeled the transducer for the communication problem as a Markov process, which provides the link to the domain of media understanding. Markov processes, as introduced in the first part, are able to represent arbitrary categorization systems as well as communication problems. Hence, the measures proposed for communication problems based on Markov processes are likewise applicable to media understanding problems. Practically, the information-theoretic analysis of media understanding stands on a sound basis. Like statistical measures, information-theoretic measures reveal fundamental properties of the media understanding system under investigation.

The central measure of information theory is *information entropy*. It is often (silently) assumed, that entropy would be a somewhat *natural* property of information systems as it is in physics. That is not the case. Shannon describes the path to a good measure of information clearly in [333]. He defines desirable properties as requirements of the process. Then, he scans the space of potential functions and eventually, decides on the *Boltzmanm H Theorem* as the best

fitting measure for his purpose. We consider this view essential to understand the potentials and limitations of information entropy. The isomorph formula for discrete information entropy e[1] goes as follows.

$$e(f) = -\sum P(f_i)\log(P(f_i)) \tag{20.9}$$

Here, f is the output of the transducer, in our case, for example, a description computed by some feature transformation. With $P(x)$ we compute the likelihood of appearance of output/description x in a sample (e.g. a feature space). The negation is necessary, because the logarithm will for probabilities always be below zero. Discussing the formula shows that the first term rises linearly while the second falls over-linearly for rising probabilities. This tension creates an interesting behavior in the evaluation process.

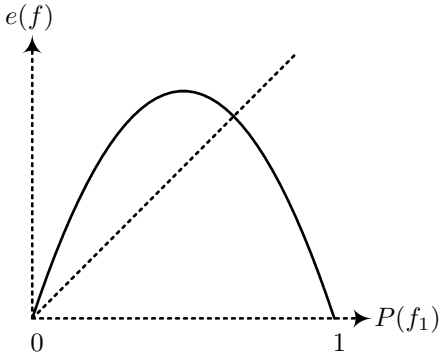


Figure 20.4: Information Entropy Example.

What is so particular about information entropy? Figure 20.4 describes the characteristics of the function for the case of $f = \{f_1, f_2\}$. Since the space of possible outputs/events/description values knows only two options, we have $P(f_2) = 1 - P(f_1)$. In this case we see that entropy approaches zero if one event is predominant. The entropy is maximal if both events appear with equal frequency. This behavior generalizes for n events, i.e. *information entropy is always maximal if all possible events occur with the same frequency.* This immediately explains why the entropy formula is tailor-made for media understanding, in particular, the analysis of media descriptions. Above, we mentioned several times that the ideal description element will – over a well-balanced ground truth – have uniform distribution. Information entropy measures exactly this property.

Information entropy assumes a mixture of ergodic sources as input. That is, all sources that create the mixed output have the same statistical properties.

[1] We stick to our nomenclature, even though h is the more common notation for information entropy.

The ergodic requirement establishes the link from information theory to dynamic systems, which can nicely be defined topologically as ergodic systems. Please refer to Chapter 27 for details on this issue. As mentioned above, we assume the source of the input to be a Markov process. In the case of $e(f)$ this Markov process is of *zero order*. That is, the appearance of one event f_i does not depend on any of its predecessors, practically $aaabbb = ababab$. If such a Markov process is not sufficient to describe the production system properly (e.g. if some concept of neighborhood exists), we require a process of higher order. A Markov process of first order (one predecessor) can be evaluated by *conditional entropy* as follows.

$$e(f) = -\sum_j \sum_i P(f_{ij}) \log(P(f_{ij})) \tag{20.10}$$

Hence, the dependency of the transducer on the past is expressed in conditional probabilities: $P(f_{ij}) = P(f_j|f_i)$, if we assume a temporal context of the indices i, j. Higher-order Markov processes can be defined in likewise fashion.

We see, theoretically, information entropy could describe the underlying characteristics expressed in some data precisely. It would only require the definition of a sufficiently high-dimensional Markov process. Practically however, this is in most instances not done. Investigators rather rely on the aura of the entropy formula and apply information entropy based on a zero-dimensional Markov process. This is straightforward but wrong. Though the optimization criterion is preserved, the structure of the underlying data is ignored. Hence, entropy measurement is reduced to the evaluation of a histogram of isolated events.

Shannon defines a number of measures based on information entropy. *Relative entropy* sets the actual entropy in relation to the theoretical maximum.

$$\text{Relative Entropy} = \frac{e(f)}{e_{max}} \tag{20.11}$$

$$\text{Redundancy} = 1 - \text{Relative Entropy} \tag{20.12}$$

$$\text{Information Gain} = e(f_j) - e(f_j|f_i = x) \tag{20.13}$$

$$\text{Negentropy} = e_{max} - e(f) \tag{20.14}$$

As mentioned above, the theoretical maximum of entropy for some description element f is the uniform distribution of all possible (quantized) values. Then, *redundancy* defines how far an actual transducer (e.g. a feature transformation) is from the theoretical maximum. For our example, if the actually created distribution of description elements is Gaussian, then we have a high redundancy, which is equivalent to the existence of neighborhood in the space of events.

The *information gain* measures the value of knowing $f_i = x$ in a conditional process by computing the contrast of the entropy over all events and the entropy

for the known event. Eventually, *negentropy* measures the free entropy, as Gibbs called it. The result is similar to redundancy but without normalization to a pre-defined range of values. Negentropy – a fashionable term in media theory, as we will see in Chapter 22 – measures the potential for improvement in the transducer. The original definition for physical applications by Schrödinger set e_{max} as the entropy of the Gaussian unit distribution, because this is the one with the highest entropy among the unit distributions. Hence, this form of negentropy measures how far a natural process is from maximal diversity.

These measures bridge the gap to the concept of *interestingness measures* as, for example, defined in [115]. An interestingness measure summarizes the *diversity* in a sequence of events.[2] The individual events are created by some process, a transducer that may, for example, be a feature transformation. In this case, the events are description values of media objects. The formulation of the interestingness measure determines its focus of attention. As we saw, information entropy judges uniform distribution of all possible events as interesting. Statistical variance is maximal for strong outliers. Some other relevant measures are listed and discussed below. See Table 11 in [115] for a comprehensive list of measures.

$$\text{Bray Measure} = \sum min(f_i, q) \tag{20.15}$$

$$\text{Gini Coefficient} = \frac{q}{2} \sum_j \sum_i |f_j - f_i| \tag{20.16}$$

$$\text{Kullback-Leibler Divergence} = -\sum f_i \log \frac{f_i}{q} \tag{20.17}$$

$$\text{Simpson Measure} = \sum f_i^2 \tag{20.18}$$

Simpson's measure is a form of energy value. Squaring the input values f_i reduces small values and increases large ones. Hence, for Simpson positive outliers are interesting. The Gini coefficient (for example, used by the United Nations to compare the state of development of countries) measures the total difference of two sets of events. The bigger the differences the higher the coefficient. Factor q is a normalizing factor (e.g. a norm in form of a query object). Bray's measure computes the sum of events where a minimum is given in form of a norm q. Hence, it is mostly interested in large values f_i. Eventually, the Kullback Leibler divergence (KLD, Q16 in the appendix) is a measure related to information entropy. It relates individual events to some norm and applies weighting afterwards. Hence, a sequence of events will be the more interesting

[2] Sometimes, in particular for text and bioinformation, the term *informativeness measure* is used as a synonym.

the more different it is from the norm. The divergence measures Q10, Q17-Q19 in Appendix B.1 can be used in the same way as the KLD.

The KLD brings us to the problem of practical application of information entropy and other interestingness measures. Firstly, the inputs f_i in KLD and the other measures have – in the context of media understanding – to be understood as probabilities $P(f_i)$. Hence, we have to aggregate the likelihood of occurrence of some event f_i over a given sample. But this is usually not sufficient. Media understanding relies mostly on quantities (e.g. quantitative color descriptions). Should we consider every two values of some description element f_i as individual events? That will hardly make sense. The practical approach is to *quantize* ranges of values into events prior to probabilistic aggregation. In simple words, compute a normalized histogram of the data before evaluation.

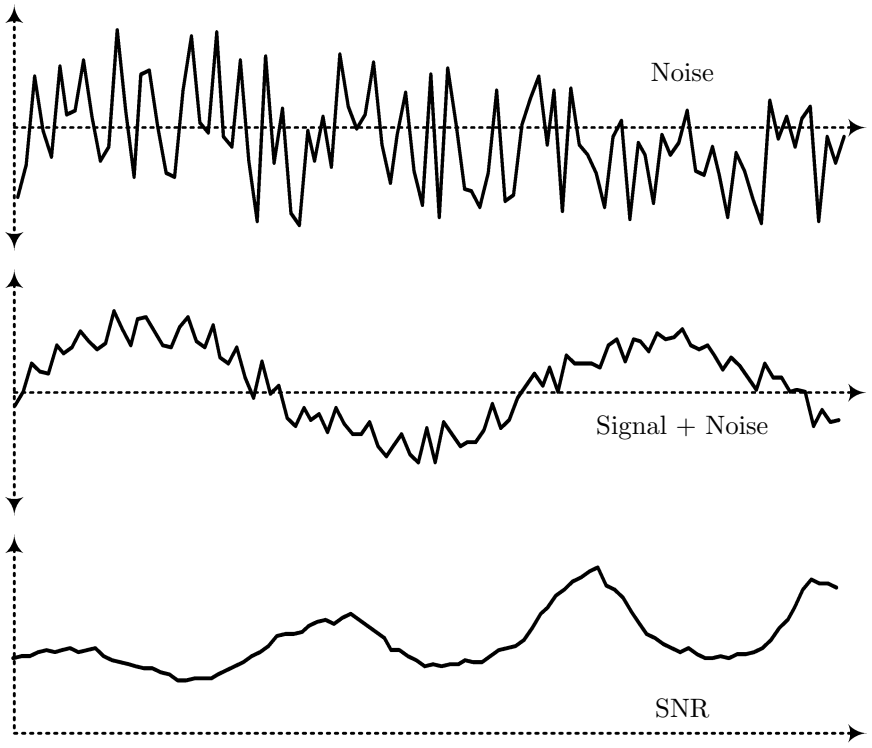


Figure 20.5: Signal-Noise Ratio Example.

Before we conclude, we would like to turn the attention once more to the *signal-noise ratio* (SNR), which is another fundamental quality criterion in me-

dia understanding that may be interpreted as an interestingness measure. The following equation gives two alternative definitions.

$$\text{SNR} = \frac{\text{Signal}}{\text{Noise}} = \frac{\mu}{\sigma} \tag{20.19}$$

Figure 20.5 illustrates a typical application for the first form. Known signal information (a sine wave, not given in the figure) is overlaid with noise. The resulting signal can be evaluated by SNR. The bottom graph in the figure shows that the SNR is maximal where the influence of the noise is low and vice versa.

The second SNR formulation is also an appealing interestingness measure. It normalizes the mean over the standard deviation. Since the mean is sensitive to magnitude and outliers (i.e. large values are interesting) and the standard deviation serves as doubt in the reliability of the data (i.e. a belief score), the result is a measure that should lie somewhere in-between statistical variance and information entropy.

We conclude that the triplet *entropy, SNR and statistical variance* provides an interesting systematic overview over any data set. Of course, the measures are partially conflicting, i.e. they cannot all be maximal simultaneously. This is equally true for the other interestingness measures. The crux is to define the role of some data set and to derive the interestingness measures accordingly. In the last section of the chapter, we do this for media understanding feature spaces.

20.4 Evaluation of Good Feature Transforms

This section focusses on descriptions. Evaluation is the central topic but we also put together some threads that were started earlier in the second part of this book. The goal is to define what an interesting feature space (the aggregation of descriptions) should look like, how this interestingness could be measured and – based on the measurement – be optimized.

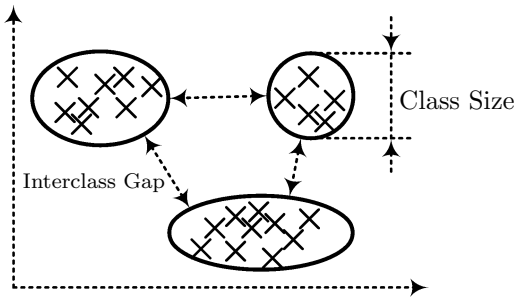


Figure 20.6: Properties of Good Descriptions.

The properties of good feature transformations were already discussed in the second section of Chapter 11. Based on the resulting feature space and some ground truth, we identified two paramount properties that are illustrated in Figure 20.6. One x stands for one description, i.e. media object, while one ellipse stands for one class, i.e. ground truth group. The members of one group should be as close to each other as possible. We called that *stability* in Chapter 11. Groups, e.g. represented by their means, should be as far from each other as possible. We called that the *discrimination* ability. Ideally, groups should not overlap and the variance of group members should approach zero. However, as we saw in the information filtering chapters this is hardly ever the case. In the ideal case, a simple decision rule would be sufficient to categorize the data. In fact, a quantization function would also do.

In Chapter 18 we already encountered an approach to transform arbitrary feature spaces in a better form: *kernel functions*. The purpose of a kernel function is to introduce space between the members of different ground truth groups. Then, so the reasoning, a simple classifier (for example, a hyperplane in the case of the support vector machine) is sufficient to categorize the description data. Unfortunately, most state-of-the-art kernel functions are rather inflexible. They neglect the stability criterion and apply the same transformation on all samples in order to improve the discrimination ability. A more flexible approach would be desirable.

In this section, we develop this approach. In order to optimize both discrimination and stability we require an interestingness measure that takes both properties into account. We start our search at the linear discriminant analysis criterion (LDA) introduced in Chapter 18. This measure has to be optimized in an iterative process. Our process is based on canonical correlation analysis, since this approach uses an advanced form of LDA. Eventually, we refactor the entire approach and come up with a novel solution for feature space evaluation and optimization.

Before we start, however, one question needs to be answered. How can it be that there is still undesired variance (e.g. in form of a lack of stability) in the descriptions after information filtering? The answer is, of course, that the information filters presented so far are all *systematic* methods, i.e. they do not take world information into account. The paramount difference of the problem discussed here is that a good feature transformation is one that represents the ground truth by stability and discrimination to an extent that categorization can be reduced to simple mapping from class means to semantic names.

We already came across a measure that takes both stability and discrimination ability into account. In Chapter 18 we wrote Fisher's LDA SNR measure for a feature space with two ground truth groups as follows.

$$\mathrm{SNR}_{\mathrm{LDA}} = \frac{m(\mu_1, \mu_2)}{\sum_{i,j} m(\mu_i, f_{ij})} \rightarrow \max \tag{20.20}$$

Here, μ_i is the mean vector of one ground truth group i over all members. The numerator is a measure of discrimination, since the coefficient is maximal if the distance m between the two means is maximal. The denominator, on the other hand, is sensitive for the stability criterion. The sum of distances from the means to the individual group members f_{ij} needs to be minimal in order to meet the global maximization goal.

The LDA is an interesting starting point for our algorithm. However, it has two serious shortcomings: Firstly, it is limited to the case of just two ground truth groups. Secondly, it is only an SNR measure that does not provide a suggestion for improvement of the evaluated descriptions. The *Canonical Correlation Analysis* (CCA) overcomes at least the second shortcoming. CCA defines the following evaluation measure and optimization criterion.

$$\mathrm{SNR}_{\mathrm{CCA}} = \frac{x\chi_{ij}y}{\sqrt{x\chi_{ii}x.y\chi_{jj}y}} \rightarrow \max_{x,y} \tag{20.21}$$

Here, χ_{ij} is the covariance of description vectors f_i, f_j, hence, χ_{ii} is the variance of one description vector. The variables x, y are weight vectors that need to be set in a way that optimizes the criterion. Please note that, inverse to LDA, the numerator measures the absence of discrimination ability while the denominator measures the absence of stability. This behavior is due to the substitution of the distance measure by the covariance/variance. The optimization of the CCA is usually performed under the constraint that the weighted variances have unit size, i.e.

$$x\chi_{ii}x = y\chi_{jj}y = 1 \tag{20.22}$$

The solution of CCA optimization is obtained by incorporation of the constraints in the goal function by Lagrange multipliers, algebraic simplification, computation of the Eigenvectors and Eigenvalues and usage of these to set x, y. The Lagrange approach and Eigenvalue decomposition fit together naturally for this type of problem. Consider the following signatures.

$$Ax = \lambda x \tag{20.23}$$

$$\frac{df(x)}{dx} - \lambda\frac{dg(x)}{dx} = 0 \tag{20.24}$$

$$\Rightarrow \frac{df(x)}{dx} = \lambda\frac{dg(x)}{dx} \tag{20.25}$$

The first line is the Eigenvector problem for a matrix A, Eigenvalue λ and Eigenvector x. The second/third line is the Lagrange approach for a goal function f, Lagrange multiplier λ and constraint g(the optimum lies where the first derivate is zero). The signatures are under certain conditions equivalent. The functions f, g given appropriately, the Lagrange multipliers can be obtained by Eigenvalue decomposition.

CCA does not provide a solution for more than a pair of description vectors. Hence, the algorithm has to be computed for each pair of vectors in feature space, which requires a form of merging (alignment, smoothing) of the individual solutions. It makes sense to embed CCA computation in an iterative dynamic process that uses the input of one iteration for refinement in the next iteration. One such transducer would be the *expectation maximization* approach.

Practical implementation and evaluation of this approach showed the author that the results are suboptimal. It turned out that the CCA optimization criterion is not strict enough for the purpose of evaluation of feature transformations. Hence, we developed a stricter procedure that performs the following preparatory steps.

1. For each ground truth group, compute the mean vector μ_i and the standard deviation over all members σ_i.

2. Sort the groups in ascending order by their distance to the origin of the coordinate system.

3. Use the vector with maximal distance to normalize all mean vectors. After this step, the most distant vector has distance 1.

4. Define a scale from the origin of the coordinate system to the mean with the largest distance and define points along this scale with constant step width. The number of points has to match the number of ground truth groups.

The result of the process is – for one-dimensional descriptions – illustrated in Figure 20.7. If we have just two ground truth groups (bottom), two points are defined along the scale (denoted by x). With increasing number of classes the reference points get distributed linearly over the space. Now, this scale is used as follows to measure stability and discrimination.

$$m_d = 1 - \frac{\sum |\bar{\mu}_i - r_i|}{n} \rightarrow 1 \tag{20.26}$$

$$m_s = 1 - \frac{\sum \sigma_i}{n} \rightarrow 1 \tag{20.27}$$

Measure m_d measures discrimination, where $\bar{\mu}_i$ is the i-th group mean after sorting and r_i is the i-th entry on the measurement scale. The black horizontal line segments in the figure illustrate the individual associations. Measure m_s expresses the average distance of the mean vectors from uniform distribution. Taking the inverse shifts the optimum to unit size.

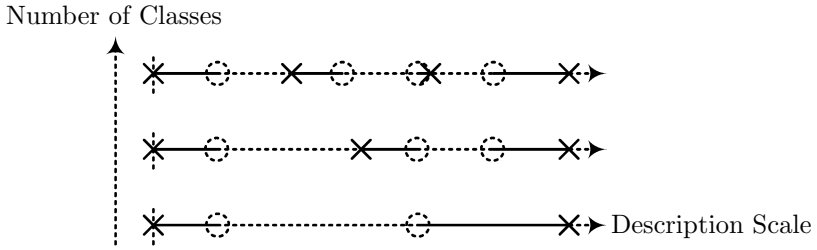


Figure 20.7: Measure for Good Feature Transformations.

Like in LDA/CCA, m_d measures stability as inverse variance. Since we are interested in well-balanced optimization, we use the F_1 score to merge the two optimization criteria.

$$\mathrm{SNR}_{ds} = \frac{2 m_d m_s}{m_d + m_s} \rightarrow 1 \tag{20.28}$$

Practical evaluation of this measure showed its superiority over the LDA/CCA approach. The used discrimination measure is significantly stricter than just distance/covariance and the F_1 score guarantees balanced optimization. However, the measure does not yet suggest an optimization algorithm. We found the following GA-like approach well-performing.

1. Compute SNR_{ds} for each description element of feature space.
2. Select the n best description elements for refinement and eliminate the rest from feature space.
3. Compute new description elements by pairwise division, multiplication, subtraction and summarization of feature space members.
4. Add the computed description elements to feature space and return to the first step.

This process is iterated until a predefined level of SNR_{ds} is reached. Experiments showed that this process increases the data quality in terms of SNR_{ds} by up to 30% which causes a significant increase in categorization performance of the corresponding media understanding algorithm.

Please observe two points. Firstly, this algorithm may be interpreted as a form of feature selection (see Chapter 20). Secondly, the suggested algorithm is in fact a kernel function. Ideally, it reaches the desired goal and reduces categorization to a labeling problem by eliminating overlaps in feature space between ground truth groups. The major disadvantage of the approach is its complexity. The improvement process for description elements takes far too long in order to be computed on the fly. One solution to this problem is to identify a successful recombination pattern during classifier training (third step of the algorithm) and to apply this kernel function statically during online categorization. The results are worth the effort.

In conclusion of this chapter and the second part of this book, we have introduced a number of high-performance algorithms for feature transformation, information filtering and categorization. Where possible, we endeavored to identify parallels between methods used in different areas. In the next chapter, we reflect these patterns, draw conclusions from the first part and sketch the objectives of the third part.

Part III

Frontiers of Media Understanding

Chapter 21

Reflection of Professional Methods

Lists the major findings of the second part, names major potentials of the professional methods, develops a set of categorization building blocks, sketches best combinations of media understanding methods and provides an overview over the third part.

21.1 Conclusions from Advanced Methods

This chapter serves the same purpose as Chapter 11: Transgression from one layer of understanding to the next. We summarize the major findings of the second part in the first section. Then, we do for categorization what we did for feature transformation before: We distill the major building blocks common to the discussed classifiers. We have seen that the world of categorization is populated by quite diverse creatures. Standardizing these is a non-trivial task though worth the effort, since it provides valuable insights on the general possibilities of categorization as well as on potentially interesting new combinations of successful components. In Section 21.3, we take what we have learnt about the big picture of media understanding and about the toolbox of feature transformations and categorization algorithms. With this input, we discuss *combinations of methods* that have proven successful in practice. In the last section, we provide an overview over the last part of this textbook, in which we move towards the frontiers of active research in the various disciplines united under the media understanding umbrella. In short, in this chapter we reflect the second part and

derive conclusions as input for the third.

As the first section of Chapter 11, this one is organized along the big picture. First, we summarize the essential points about professional feature extraction methods and information filtering methods. Then, we emphasize the major findings about categorization and evaluation. Eventually, we point out the – from our perspective – major shortcomings of the state-of-the-art media understanding methods and discuss starting points for improvement.

In the area of feature transformation and general information filtering we see five paramount aspects of professional media understanding methods.

1. *Template matching* based on *discrete transforms* is currently one of the most effective approaches to improve the semantic level of descriptions. The set of discrete transforms ranges from the straightforward Fourier transform – which is still state-of-the-art in the audio domain – to multi-dimensional wavelets. The transformation step provides a global convolution operation that is most successful where the content matches the template provided by the transform perfectly. Basing template matching on the spectral representation allows for pushing the semantic level even further up. Here, the template can be any form of representation of context, as we defined it in Chapter 11. It is one challenge of future media understanding research to identify those types of context that influence human perception and similarity judgment profoundly. Since this topic is of eminent importance, two chapters of the third part deal with it. In Chapter 24 we deal with the problem of semantic template matching in general and in Chapter 25 we discuss current best practices.

2. The *identification of local interest points* based on the characteristics of their gradients and the *usage of sets of these interest points* to describe objects is a second major approach that improves the semantic level of descriptions significantly. The idea is motivated by insights in human perception dating back to the 1950ies. The state-of-the-art approach includes the computation of a scale space before interest point detection and neighborhood-based description afterwards. In particular, the second idea leads to expressive local descriptions. As we will see below, there is some potential in the actual selection of interest points. We saw that the currently used approaches are generally similar and mostly perform selection based on high curvature. Visual investigation, however, shows that the interest points selected by these algorithms do not preserve the visual characteristics to a satisfactory degree. Still, the general approach has the highest potential to be the foundation of a future human-like object representation method for media understanding.

3. In the 1990ies, some leading researchers considered the transformation-

based approach of *description by wavelet-based multi-resolution analysis* as the ideal description of media objects. This form of transformation should include all relevant information, if the mother wavelet fits *both* the characteristics of the media (e.g. edge information in images) and the characteristics of the desired information (e.g. human faces). Unfortunately, as we discuss below, twenty years later still very simple mother wavelets are mostly used for transformation. On the other hand, the *local feature extraction approach* in combination with scale spaces provides a very similar result to wavelet-based multi-resolution analysis. We discussed this issue in Chapter 14. It is, therefore, thinkable that the future will see further convergence of these fundamental approaches and mutual fertilization.

4. In the domain of description of temporal change, we have seen that *optical flow*, the method of choice, is the basis of most semantic descriptions. Independent of the size of the media sample for which the flow vectors are computed (one pixel, macroblocks, objects or even an entire frame), the general algorithm includes neighborhood search (a form of autocorrelation) and smoothing. The neighborhood search part is very similar to other methods of gradient computation. The major difference to the local methods discussed in the second part of the book and the texture descriptions discussed in the first one is that here, the gradient is computed over time. Hence, the individual flow vector is a measure of movement, and the aggregation a measure of motion activity. Depending on the context, this activity can be interpreted in several forms, for example, as camera motion or the movement of complex objects.

5. Eventually, the hidden message of the information filtering methods introduced in the second part is that descriptions – no matter how intelligently extracted – are not sacred. Advanced filtering methods help to reduce the level of redundancy, make the descriptions more efficient (i.e. they express more information with less numbers) and to *smooth out noise*. In particular, the latter point provides a link back to the last conclusion, where smoothing is the essential step in aligning neighboring flow vectors. For example, the Lucas Kanade approach adopts regression for this purpose. It is a general property of media understanding approaches that smoothing methods are used to fit the sample data to similar smooth functions. The advantage of increased efficiency is, however, paid with the potential *loss of noise-like characteristics that are relevant for human perception.* Humans are not machines. Despite all our efforts to create conflict-free mental models of our concepts, our experience and cognition are often not smooth. Media understanding has to endeavor to imitate human cognition in order to be successful. In the third part, we will discuss this issue intensively in several chapters.

In the area of categorization and the evaluation of training and application processes, we would like to stress the following major aspects of the methods discussed in the second part of the book.

1. Human learning and machine learning are highly related – beyond the names given to machine learning methods. The discussion of concept theories showed us that *philosophical reasoning has come up with the same two fundamental approaches as machine learning.* The classic one is based on constraints, we call such methods separators. The other is based on examples (prototypes), we call such methods hedgers. Furthermore, we have seen that some methods are somewhere between hedging and separation, for example if the separating micro process is used in a learning macro process that transforms it into hedging. The existence of such methods points in the same direction as the most recent developments of concept theory: Most concepts may be mental theories under lifelong development. That is, an initial hedger or separator is enhanced by exceptions and limitations which makes it less smooth than it was before. What we know as maturing by experience can be seen as an inverse process to the smoothing by information filtering advocated in the last point. Maybe the future of media understanding lies in giving up model rigidity for better adaptation to human behavior.

2. The *structural risk minimization principle* makes the fundamental goal of rigid machine learning (in the sense of the last point) explicit. Categorization knows two goals: the minimization of the number of misclassifications and the minimization of the computational effort (time, resources). The first goal has been considered most important in the past. Loss functions have been defined for the three major categorization problems: classification, regression, density estimation. It was natural to search for the algorithm that performs these operations best and to neglect performance considerations as long as no satisfactory solution has been identified. However, machine learning has advanced to a state where almost arbitrary patterns can be learnt. Hence, the minimization of complexity becomes relevant for two reasons. Firstly, performance is a practically important issue. Secondly, the simplification of the categorization model helps to minimize the danger of overfitting, which is – in the absence of generally accepted ground truth for most categories (Plato's problem!) – a practically highly relevant issue. Perfect representation of a training set is unsatisfactory if the set is unsatisfactory.

3. Simple models help avoiding overfitting but increase the chance of a loss. The congenial partner of structural risk minimization that helps overcoming this catch-22 is adapting the descriptions to the needs of the simple

categorization function. *Kernel functions* are the methods that provide this functionality. In the second part, we saw that kernel functions have been defined for almost all types of media data. Their success lies in two strategies: blowing up the dimensionality of feature space while keeping its population constant, and template-like reorganization of feature space. Hence, the approach links back both to information filtering and template-based matching. The major disadvantage of the kernel methods of today is that they are mostly static. That is, the same few kernel functions are applied on all kinds of data. In the next list, we discuss a method with higher potential. Still, the big merit of kernel-based methods is their amalgamation of feature extraction, information filtering and categorization methods. A perfect kernel would reduce the latter problem to straightforward labeling.

4. Psychological findings suggest the *representation of concepts by norms, i.e. mixtures of density functions* that represent human experience. Such a norm would be a suitable implementation of the *theory theory* mentioned above (concepts are work in progress). In machine learning, several methods exist for the creation, refinement and application of mixtures. We have seen that, for example, the Gaussian mixture model is a simple yet effective classifier that can as well be used for the definition of the confusion matrices of Bayesian methods. The important point here is that this approach fits human cognition well, which is after all the yardstick of all media understanding efforts. Furthermore, this approach as well as the conceptually similar boosting methods provide a link to the theory of dynamic systems which will be discussed in the third part of the book.

5. The evaluation process can be simplified significantly by the introduction of standardized procedures and measures. For this purpose, we introduced *cross validation and the receiver operating characteristic curve*. Combining these two methods reduces ground truth-based evaluation to a visual control mechanism. Furthermore, standardization helps the exchange of results. It is, therefore, highly recommendable to employ these two methods for the evaluation of media understanding schemes.

We would like to close this section by illuminating five major shortcomings of the methods employed today. The selection is subjective, of course. We consider the listed issues interesting holes in the theory that represent potentials for future development.

1. So far, hardly any *n-dimensional semantic wavelet mother functions* have been defined. Above, we named faces as one example. Figure 21.1 illustrates schematically what a wavelet bank of faces could look like. The

advantage is obvious. Convoluting a media source over a bank of semantically relevant objects leads to a multi-dimensional representation that is also semantically relevant. The disadvantage, on the other hand, is also easy to see. The number of dimensions that a semantically relevant wavelet bank would have to have is significantly larger than in the one-dimensional standard case. There, we have just location and scale. A face wavelet base would require aspect ratio, expressions, presence/absence of hair, etc. In short, improving the semantic gap is bought by worsening the curse of dimensionality. Still, semantically relevant media understanding must necessarily follow this path. The dimensionality problem can be reduced by carefully investigating those aspects (and correlations of aspects) of the semantic categories that are really relevant for human cognition.

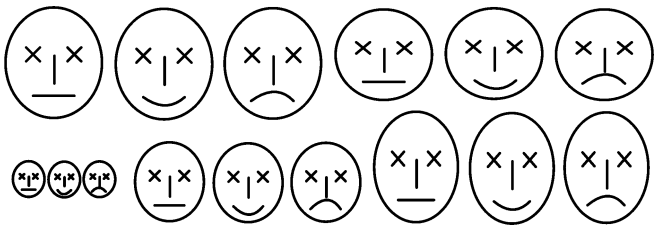


Figure 21.1: Semantic Face Wavelet Example.

2. *Local interest point detectors* – though motivated by cognition – *do not take human perception into account* (e.g. Gestalt rules). Figure 21.2 shows an example. The left edge map from the leading example is represented twice by interest point detectors and once by a human. We see that the Harris corner detector is hardly able to represent the relevant face information. The most important face features (e.g. the eyes) are almost completely lost. The performance of the Laplacian of Gaussian operator (LoG) is slightly better: Some eye and mouth features are preserved. A human test person, however, would distribute the same amount of local information differently over to stimulus. The rightmost image preserves the features of the original stimulus well but does not produce a longer description than the LoG approach. This is similarly true for most types of objects (except simple geometric primitives). We conclude that curvature is an important selection criterion, but not the only one. Periodic selection of feature points (as in the Gestalt laws) is important as well. Of paramount relevance, though, is the preservation of the semantically relevant properties (eyes, nose, mouth, etc.). Hence, for media understanding more effort should be invested in the analysis of context/semantics and their tailor-made representation in feature transformations.

Figure 21.2: Do Machines Select Interest Points Like Humans? (© *CNBC*)

3. *Optical flow should be based on real-world objects.* That is, flow computation should follow the recognition of object contours or of local feature sets. It would be straightforward to base flow computation and neighborhood search on selected interest points. The reduction of the number of points to investigate would even compensate for the additional effort. Figure 21.3 shows a practical example. If the face contour of the anchor person was extracted before the flow computation, the entire process could be limited to the features of the face, which would increase the performance, reduce the amount of noise and, therefore, make smoothing obsolete. The computation of object-based flow is currently a hot topic in computer vision. We are positive that the ongoing efforts will improve the semantic relevance of motion features significantly.

Figure 21.3: Object-Based Optical Flow Example.

4. Psychological knowledge about the nature of *human similarity judgment and generalization is not sufficiently used in the micro process* of categorization. The generalization curve is the answer to the general problem of how long differences appear similar. This knowledge is fundamental for the correct judgment of category memberships. Human judgment is

not linear like a k-nearest neighbor classifier. If the distance between two stimuli is too large, they are considered different no matter how sparsely populated the space of examples is. Since human similarity judgment is a highly relevant source of information, the entire Chapter 28 is dedicated to the discussion of how this knowledge can be employed for machine categorization.

5. Eventually, above we criticized that general-purpose kernel functions are a major step forward but not the last word on the subject. We sketched an advanced approach that optimizes the kernel function based on the given descriptions and ground truth. Using a simple classifier in combination with such a learnt kernel is very similar to the boosting approach. The tunable kernel learning approach introduces a continuum of possible classifiers ranging from boosting to structural risk minimization. The complexity of the categorization problem is shifted from similarity measurement to model building. We believe that this tailor-made kernel functions are suitable for the representation of complex concepts/norms.

In conclusion, the complexity of the media understanding problem can be shifted between different points of the big picture. We can either try to extract semantically relevant descriptions, or we use the semantic knowledge to build an intelligent classifier, or we use a simple classifier on simple descriptions and build a semantic kernel that maps simple descriptions on a semantically enriched space. In the third section, we investigate successful combinations of these media understanding methods. Before that, however, we anatomize the categorization process.

21.2 Building Blocks of Categorization

Understanding the building blocks of a method is necessary in order to understand its functionality. We consider it beneficial to analyze the building blocks of categorization – as we did for feature transformations –, because the set of machine learning methods appears highly diverse. Identifying the building blocks will show that most differences are not as big as they appear. Our approach takes the following path. First, we organize the set of methods by the required training data and by the categorization principle. The discussion of the emerging fundamental types leads to preliminary sets of components which are, then, unified in a general set of categorization building blocks. This set is made subject to analysis with respect to the fundamental problems of media understanding. Eventually, we compare the building blocks of feature transformation and those of categorization. Above, we argued that the complexity of the media understanding process can be shifted between the components. This suggests

that similarities between the building blocks should exist. The last part of the section is dedicated to the identification of these similarities.

Required Context	*Hedgers*	*Separators*
None	Cluster Analysis, Self-Organizing Map	Random Selection
References	K-Means Vector Space Model,	Decision Tree
Ground Truth	Gaussian Mixture Model, K-Nearest Neighbor	Artificial Neural Nets, Bayes Nets, LDA, Support Vector Machine

Table 21.1: A Categorization of Categorization Methods.

Table 21.1 categorizes the already introduced categorization methods according to the required training data (none, references, ground truth) and according to the categorization model (hedgers, separators). Cluster analysis and the self-organizing map are two typical hedgers that do not require any world information. In comparison, random selection is just a dummy for no learning at all. If no training data is available, reasonable separation is not possible.

If references are available, strong methods are available in both categories. Vector space model and k-means differ mostly in the application (retrieval vs. browsing). Decision trees may also be seen as intermediates between hedging and separation – depending on the number of weak classifiers employed.

The presence of ground truth information enables the usage of even more powerful learning algorithms. K-nearest neighbor is the typical hedger, while mixture models may also be seen as intermediates. The Gaussian approach will certainly try to hedge members of the same class. However, the norms may become complex enough to interpret them as separation rules. On the other hand, perceptron-like nets, probabilistic nets and the support vector machine are rather separators than hedgers. While the classification is undisputed for the support vector machine, certain neural nets (e.g. radial basis functions, see Chapter 29) may also be seen as intermediates or even as hedgers.

Before we continue with a detailed analysis of hedgers and separators, we have to point out that some methods are missing in Table 21.1, in particular, boosting methods. These form, together with decision trees and random forests, the group of *ensemble methods*. These methods appear generally similar to intermediates, since they have both characteristics of hedgers and separators. Most ensemble methods employ a multitude of simple separators to construct a categorization process sufficiently complex to be called a hedger. We consider it superior to classify boosting as an intermediate rather than a separator.

What are the typical steps performed in a hedger? The most important step in hedging is certainly *similarity measurement* from a reference to some

other object. The classification depends heavily on the measurement process. The result of measurement, though, is heavily influenced by various forms of *quantization* that are used in hedgers (for example, decision rules, thresholds, neglecting certain description elements). Some hedgers perform a *learning procedure* though the majority of the methods is static (e.g. reference adaptation in the self-organizing map vs. static k-means). Eventually, the categorization model is usually only of minor importance (e.g. references are chosen randomly in the self-organizing map).

Separators, on the other hand, invest the largest effort into *model estimation and learning*. In order to make the best of the world information, complex categorization models are constituted and refined/learnt over time. *Quantization* is of minor relevance but still used in separators (e.g. thresholds in decision rules). *Similarity measurement*, the classic micro process, is of even less importance. If the model is on a high semantic level, similarity measurement may even become obsolete. Then, it can be substituted by a quantization function.

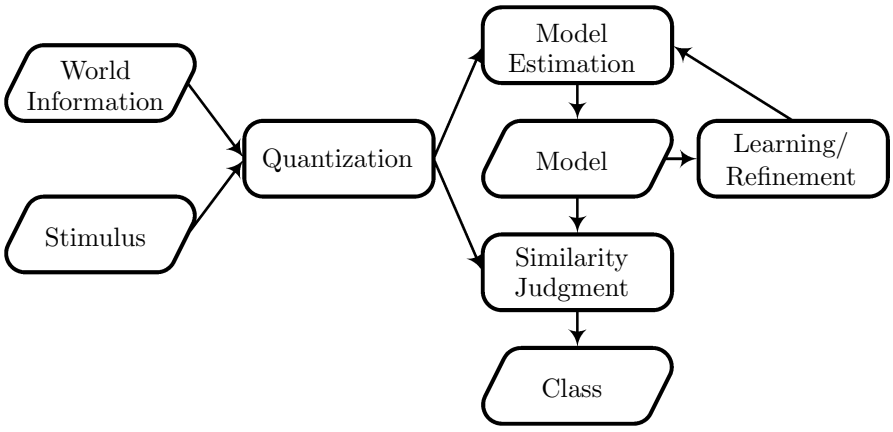


Figure 21.4: Building Blocks of Categorization.

In summary, as already sketched in Chapter 17 we come up with the four major building blocks of categorization illustrated in Figure 21.4. Quantization is every operation that transforms the input data without an attempt to relate it to other media objects or to aggregate over description space. One example is the application of decision rules in decision stumps. Model estimation is the process that builds the categorization model from the (quantized) input data. A typical example is Gibbs sampling for Bayesian models. The learning process (control loop, refinement process) controls the training process. Here, a typical example is the expectation maximization algorithm. Eventually, similarity judgment is the process that relates two (quantized) descriptions by a form of crosscorrelation.

The classic example is distance measurement of query and feature space members in the vector space model.

The four building blocks are formalized as functions in Appendix A.6. We use $estimate_i$ for model estimation, $learn_i$ for the learning/refinement process, $measure_i$ for similarity measurement, and $quant_i$ for quantization.

If we join the list of building blocks with the categorization in Table 21.1 we can derive a number of interesting conclusions. Firstly, hedgers make hardly any use of the value of ground truth information. Most models are relatively simple. A notable exception is the Gaussian mixture model – another reason to regard it rather as an intermediate. The simpler models reduce, of course, the influence of the (incomplete) ground truth and the danger of overfitting. On the other hand, the complexity of the categorization process lies in the actual classification. Separators shift the complexity generally to the training process, which allows fast application. The complexity of most similarity measurement procedures increases the gap in execution performance between separators and hedgers even further.

Property	*estimate*	*learn*	*measure*	*quant*
Efficiency	+	−	−	+
Generalization	−	−	+	+
Independence	−	−	+	+
Performance	+	+	+	−
Simplicity	−	−	+	+

Table 21.2: Influence of Categorization Building Blocks on Good Categorization.

Now that we have a set of building blocks, it would be interesting to see how they influence good categorization. A good categorization method is distinguished by five properties.

1. *Performance.* Ideally, the loss should be zero and the evaluation score maximal.

2. *Generalization.* There should be no tendency towards overfitting nor underfitting.

3. *Efficiency.* The computational requirements of training and application should be minimal.

4. *Simplicity.* The categorization model should be as simple as possible.

5. *Independence.* As little as possible world knowledge should be required for the training process.

Table 21.2 summarizes or findings about the influence of the building blocks on these requirements. Model estimation should improve efficiency and performance – the latter because algorithmic complexity is shifted from application (more frequent) to training (less frequent). The other requirements will rather suffer under a complex model. The learning process optimizes performance which has to be paid by a negative influence on all other factors. Similarity measurement has a positive influence on all requirements except efficiency, since similarity functions are usually complex and need to be computed during application. Quantization, eventually, has a positive influence on all requirements except performance, since here we give up preciseness for efficient computation.

Issue	*estimate*	*learn*	*measure*	*quant*
Curse of Dimensionality		–	–	+
Incomplete Ground Truth	–	–		+
Noise and Missing Data	–	+	–	+
Performance		–	–	+
Polysemy	+		+	–
Semantic Gap	+	+	+	–

Table 21.3: Influence of Categorization Building Blocks on Fundamental Media Understanding Problems.

What is the influence of the building blocks of categorization on the fundamental issues of media understanding? Table 21.3 summarizes the most important findings. Quantization has a positive influence on all performance-related issues. The downside is a negative effect on the handling of polysemy (due to the tendency to reduce all noise-like components) and on the semantic gap. Quantized descriptions are likely to be on a semantically lower level. Similarity measurement buys intelligent handling of context-related issues with relatively bad performance and sensitivity to the existence of noise. The learning process helps to reduce the semantic gap but is time- and resource-consuming. There is a tendency that learning will compensate for missing data and occlusions. Eventually, model estimation should endeavor to represent the semantics of the media understanding problem in the categorization model. However, the process is sensible to missing data and noise, as we discussed in various chapters of the second part.

We would like to close this section with a brief discussion of similarities between building blocks for feature transformation and categorization. Such analogies do exist, as Figure 21.5 shows. Quantization is more or less the same in both stages of the process. It is true that some quantization methods (e.g. coarse representation) are rather typical for feature transformation while others

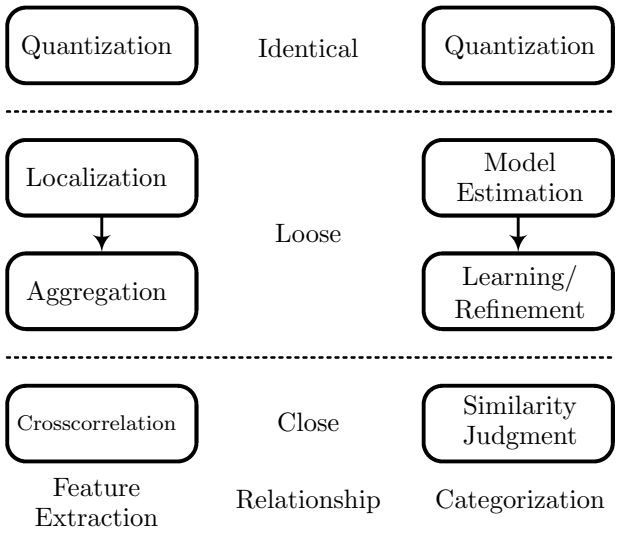


Figure 21.5: Complete Set of Building Blocks.

(decision rules, thresholds) are typical for categorization. Technically though, the processes are almost identical. There is, furthermore, a loose relationship between the localization/aggregation sequence and the estimation/learning process. Both processes construct models bottom-up. The major difference is the feedback loop which is crucial in categorization but does not exist on the feature transformation side (yet). The similarity is much higher between crosscorrelation and similarity measurement. In fact, the two building blocks comprise almost the same sets of methods, though in the latter we have in addition some methods from the domain of human similarity judgment. In summary, the similarity between the methods is surprisingly high. Categorization may be seen as a cyclic feature transformation process that ends up in very short descriptions, the class labels.

Today, there are hardly any large holes in the set of categorization methods. Ensemble methods and intelligent kernels allow to exchange one method for the other gradually. The decision over model rigidity, fit to the ground truth, learning curve, generalization behavior and algorithmic complexity lies almost completely in the hands of the experimenter. The remaining question now is: Which classifier for which descriptions? This question is discussed in the next section.

21.3 Which Methods When?

So far in this book, we presented methods that are relevant in media understanding, analyzed them and compared them against each other. Methods that operate on the same data type or perform similar operations were grouped together. In this section, we go one step further and suggest useful (combinations of) methods for particular data types, description types and classification problems. The presented list of best practices cannot be exhaustive. Rather, we sketch a framework for a future more detailed investigation of the subject.

The motivation is straightforward. We have established a process of media understanding that works for all data types under consideration and we have built a toolbox of methods for description, filtering, categorization and evaluation. The next logical step is the definition of templates of concrete media understanding processes. Or presentation follows the big picture, i.e. we deal with feature transformation first, then filtering, categorization and, eventually, evaluation.

Media Type	*Strongest Feature Transformations*
Audio	Mel-Frequency Coefficients, Perceptual Linear Prediction
Bioinformation	Start and Stop Codons
Biosignals	Correlogram
Image	Scale-Invariant Feature Transform, Color Histogram
Text	Bags of Words of N-Grams
Stock	Zero Crossings Rate
Video	Lucas-Kanade Optical Flow

Table 21.4: Best Descriptions per Media Type.

Some feature transformations are of undisputedly good performance. For example, it is advisable for any audio understanding application to include the mel-frequency cepstral coefficients (MFCC). Local visual information can always be described well by interest points. Optical flow is a good foundation for any form of motion description. Table 21.4 provides a list of best feature transformations per media type. For audio, perceptual linear prediction is a good supplement to MFCC, because the latter summarizes the signal while the former performs autocorrelation. Bioinformation processing does not really have good descriptions yet. The named codons are – though highly important – trivial properties of gene strings. The correlogram is a good basis for all relevant problems of biosignal understanding. For image data, the scale-invariant feature transform is one local interest point method that produces good results. In addition, a color histogram can be employed to describe the global image characteristics. Text can be described well by n-grams of letters and words. In combination with the

bags of words method (i.e. a histogram), strong descriptions can be computed. For stock data, we found out in experiments that the zero crossings rate has an exceptional predictive power. In combination with a simple decision rule, we could predict the development of thousands of shares over a period of two weeks correctly in more than 80 per cent of all cases. Eventually, Lucas-Kanade optical flow is a fair choice for all motion-based descriptions.

For information filtering of descriptions, it is always recommendable to employ a principal component analysis in combination with normalization and some feature selection technique. If feature merging is required, it has several advantages to perform early fusion, i.e. fusion before normalization and factor analysis.

Situation	*Classifiers*
General overview required	Cluster Analysis, Self-Organizing Map
Given concepts	K-Means
Well separable classes	Mixture Models, Support Vector Machine
High variance in class size	Bayesian Networks, Markov Processes
Many small classes	Decision Tree, K-Nearest Neighbor
Hardly separable classes	Ensemble Methods, Kernel-Based Methods

Table 21.5: Best Classifiers for Particular Application Scenarios.

Table 21.5 lists a few application scenarios (types of feature spaces) and suggested categorization methods. For a first overview, cluster analysis and the self-organizing map are the methods of choice. If the classes are given (e.g. as references), it is preferable to use the k-means approach as the straightforward implementation. If the classes are well-separable (only small overlap between concepts), density-based methods can be used as well as simple regression models such as the support vector machine with linear kernel function. If the classes are of significantly different sizes (in terms of media samples), it is advisable to employ a Bayesian method since these methods are sensitive to the greater belief expressed by greater clusters. In contrast to a support vector machine, a Markov process will lay the separation line closer to the smaller clusters thus questioning their boundaries rather than those of the larger clusters. If feature space consists of many small clusters, the k-nearest neighbor approach can be employed as well as a decision tree. Eventually, if the classes are hardly separable, it is recommended to rely on some ensemble method or to transform feature space by a non-linear kernel function.

Table 21.6 investigates a detail of the categorization process. It suggests best similarity measurement micro processes for given types of description data. Density functions can, of course, be evaluated by probabilistic inference. However, one alternative would be to employ an interestingness measure such as

Description Type	*Micro Process*
Density	Correlation, Kullback-Leibler Divergence, Probabilistic Inference
Histogram	Minkowski-Distances, Dot Product, Earth Mover's Distance, Mahalanobis Distance
Moment	Correlation, Thresholding
Predicate	Feature Contrast Model, Pattern Difference
Signature	Hausdorff Distance, Dynamic Time Warping
Symbol	Hamming Distance, Number of Co-Occurrences

Table 21.6: Best Micro Processes for Particular Data Types.

the Kullback-Leibler divergence. If the density is coarse, correlation may also be an alternative. For histograms, all kinds of distance functions can be employed. Meta-models such as the earth mover's distance may also be interesting. Moments are best evaluated by some thresholding function (e.g. mean) or by correlation/covariance (e.g. variance). Binary predicates are typically processed by predicate-based measures. The two given forms are particularly successful representatives from the list in Appendix B.2. Signatures (e.g. shape outlines, stock templates) can be evaluated by the Hausdorff distance or one of its equivalents but as well by dynamic warping. For symbols, similar to predicates, any counting measure can be used. Hamming distance has been successfully used on text, while the number of co-occurrences (the discrete equivalent of the dot product) is always a fair choice.

Eventually, if ground truth is available, evaluation can always be based on cross validation and receiver operating characteristic curves. The F_1 score is also a fair choice. If no world information is available, systematic statistical evaluation is the method of choice. The listed methods provide a fair ground for successful media understanding in the named domains. Professional media understanding is based on these methods.

21.4 Overview Over Scientific Frontiers

The organization of the third part – like the first two – follows the big picture of media understanding. Figure 21.6 illustrates the position and docking of the four major topics. We deal with practical issues such as semantic template-based correlation, dynamic behavior and the optimization of learning. On the theoretical side, we investigate what is known about human media perception, processing and comparison.

The two subsequent chapters deal with human perception. In the first, we

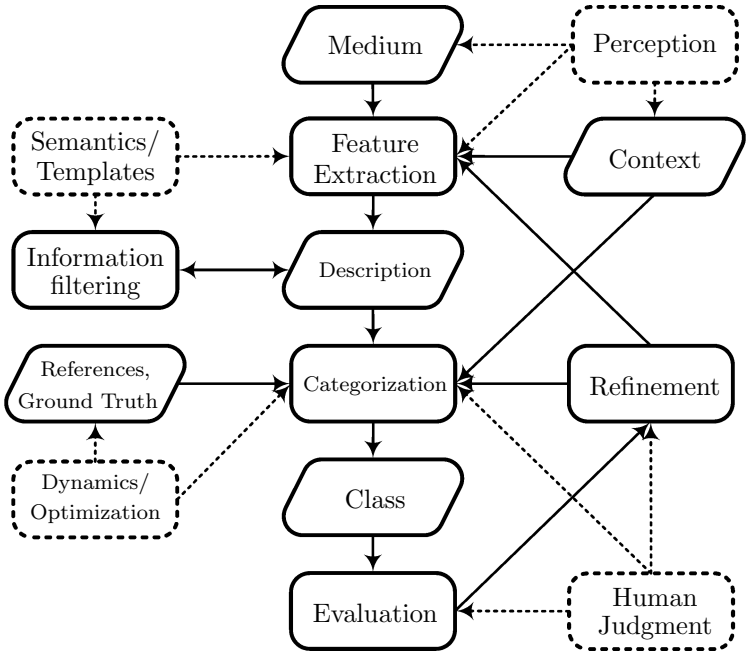


Figure 21.6: Topics of the Third Part (dotted).

collect everything that may help understanding human perception of media in general. Topics include media theory, semiotics and advanced models from information theory. The second chapter summarizes today's knowledge about psychophysics, i.e. the reception of stimuli by the individual. The already presented bits about the psychology of hearing and vision are set into context and embedded in a larger framework.

The next two chapters deal with practical frontiers of media understanding. In the first, the paramount importance of auto- and crosscorrelation for media description is emphasized. We develop a framework of currently used methods and provide an outlook into the future. In the second chapter, we tackle the semantic barrier. Based on what has already been said about context and semantics, we investigate the state-of-the-art in semantic media understanding. Application domains include face detection, speech recognition and stylometry.

After a chapter on advanced filtering and quantization methods follow two theoretical chapters of which the first deals with learning theory (macro process) and the second with human similarity judgment (micro process). We will present the state-of-the-art in the description of learning bounds and the estimation of learning parameters. Then, we introduce the four major models of

human similarity judgment and discuss, how far these models have already been implemented in media understanding. The central statement of media understanding is that *man is the measure*. A method, however good, will not be successful if it does not imitate human behavior. For example, linear regression is a fair estimation technique. From the human perspective, however, it has to be considered too perfect for representing human judgment adequately. In the third part we will introduce the term *supersemantic* for this deficiency.

Of the last two chapters, the first investigates neural models of media understanding. Starting from the perceptron, the state-of-the-art models are explained and the current frontier – fourth-generation spiking nets – is discussed. Eventually, the summary over all three parts emphasizes the major findings and points out the top potentials of media understanding research.

Chapter 22

Media Philosophies

Discusses the relationship of perception and reality, theories of media content and media usage, the semiotic analysis of arbitrary symbol systems and potentials for merging of media theory, semiotics and information theory for the benefit of better media understanding.

22.1 The Image in Philosophy

This chapter is arguably the softest of the entire book. We deal with the *media* itself – before any attempt of feature extraction or categorization. We review the most important media theories and endeavor to extract conclusions that can be employed in the scientific process of pushing the frontier of media understanding. For this purpose, the first section discusses epistemological aspects of objects and perception. The second section introduces the major media theories (for example, the Toronto school) and connects them with the technical perspective. Section 22.3 approaches the phenomenon *media* from the analytic perspective and introduces the toolbox of semiotics for the analysis of media instances. The last section provides the bridge from the world of media philosophies to the world of media understanding. This bridge is information theory. We link philosophical concepts to transduction and cognition thus operationalizing major philosophical findings for usage in engineering.

In this sense, this chapter is typical for the chapters of the third part. We deviate from the path of professional media understanding towards other research areas and collect interesting ideas that have a potential for exploitation in media understanding. The next chapter is the twin of this one as it investigates the me-

dia perception problem from the psychological and psychophysical perspective where, here, we investigate it from the philosophical perspective. The inclusion of media theory in the media understanding process is in a very early stage. So far, thoughts about the perception and cognition of media events hardly played a role in media understanding. Opening this domain for usage is a true extension of the frontiers of media understanding.

The motivation for this chapter is straightforward. It should be enlightening for our work if we understand how human beings approach the phenomenon *media*. Feature extraction and categorization have to be based on the behavior of man in order to be successful. Philosophy takes this view and analyzes media, their perception and their influence from the perspective of the human being.

This section provides the entry point for the subsequent sections. Here, we gather the major prerequisites for the understanding of media theories, semiotics and media cognition. The central topic is *epistemology*, the theory of cognition/understanding.

Since all phenomena lie outside of the human brain, every epistemology requires a handle for the representation of the world outside the brain. This handle is the *image*. The Oxford Dictionary of English Etymology defines an *image* as the *artificial representation of an object, likeness, statue; (optical) counterpart; mental representation.* This description points already at the ambiguity of the image in philosophy and psychology. Philosophical image theory defines three relationships that can make something an image of some phenomenon.

- *Syntactic*: The image has the *same properties* as the phenomenon.
- *Semantic*: The image appears *somehow similar* to the phenomenon.
- *Pragmatic*: There is a *relationship of usage* between the image and the phenomenon.

In Section 22.3, we will see that this differentiation is linked to the three possible relationships of a signifier and a signified. In fact, the relationship of image and phenomenon is a semiotic relationship. Please note that the syntactical relationship is equivalent to predicate-based similarity measurement while the semantic relationship can be seen as quantitative similarity measurement (see Chapter 28). In media theory, syntactical properties are also called *digital indicators* and semantic properties are called *analogous indicators*.

The three principal relationships of image and phenomenon have been understood for a very long time. Plato already discussed this issue and based, for example, the analogy of the cave on it. He emphasized the importance of the syntactic relationship (*eidos*) and admitted only little relevance to the semantic relationship: *Eikons* (icons), to his understanding, are only the shine but not the heart of a phenomenon.

Aristotle argued against this view by stating the holistic principle that *the whole is greater than the sum of its parts*. Hence, a set of syntactic predicates can never represent a phenomenon entirely. This view was supported by Pliny who relates the famous story of Zeuxis and Parrhasius. Both painters were able to produce photorealistic paintings. The first painted grapes that were so realistic that birds pecked after them. Parrhasius overtrumped Zeuxis by painting a curtain so realistic that it made the other want to open it. Two examples of mimesis (semantic images).

In the 1960ies and 1970ies, Goodman argued for the position of Plato by pointing out that similarity is neither a necessary nor sufficient condition for a good image. He declines the importance of the semantic relationship and rather emphasizes syntactical relationships – which is in line with the psychological point of view at that time. As we will see in Chapter 28, today, we see the ideal image as a mixture of syntactic and semantic aspects. The pragmatic aspects are due to their high semantic level (i.e. dependence on context) always out of the media understanding discussion.

Modern philosophy saw a renaissance of the image problem in Hume's epistemology. His *law of resemblance* stresses the importance of the similarity of past and future. As it cannot be proven, any form of induction/causality (cause and effect) must simply rely on the existence of this similarity. In the second part, we came across the story of the rainmaking marbles reported by Barley. This induction is a nice example for the danger in Hume's law.

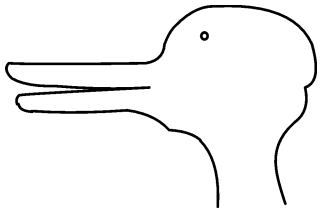


Figure 22.1: Rabbit Duck Illusion.

In the Philosophical Investigations, Wittgenstein investigated the effect of the *reversible figure* shown in Figure 22.1. Depending on the viewpoint this figure may be seen as a rabbit or a duck. The perception may flip over time. Wittgenstein called this a *change of aspect* and tried (unsuccessfully) to find its reason. Obviously, the figure shows a semantic relationship to both the sketch of a rabbit and a duck. The predominant explanation today is based on semantic relationships though. Since the figure shares an about equal number of properties with both animals (position of the beak/ears, eye, etc.), it has an about equal similarity to the human concepts of both animals (without asking if this concept is based on conditions or prototypes). Hence, both concepts race in a cognitive

process for the successful explanation of the phenomenon. Since the distance between them is small and visual perception is a continuous stream of saccadic eye movement, oscillation in the form of changes of aspect is the consequence.

What can we conclude from this discussion? Our media are the phenomena. The image is the human perception of the media. That is, perception is based on common properties, similarities of appearance and similarities of use/part-whole relationships. As we will see in the discussion below as well as in Chapter 28, any theory that wants to understand media has to operationalize these three relationships. Equipped with these philosophical tools, we advance in the field of media theories and investigate its most prominent representatives.

22.2 Media Theories

In this section, we provide a rough overview over the major media theories. We start with a set of goals for media theory. Then, we introduce the major contributions in temporal order. The first media theories were mostly concerned with aesthetic problems. Only after the second world war followed more technical theories – possibly inspired by the advances in cybernetics, dynamical systems and information theory. A major milestone were – and still are – the ideas of Marshall McLuhan, the major exponent of the Toronto school. Later decades saw the development of postmodern media theories that went again further away from the technical standpoint. One type of media has been of especial importance: language. Hence, in the last part of the section we sketch the major theories that deal with text, language and writing. Media theories are always bound to individuals and, therefore, the following discussion goes from one major exponent to the next.

Medium	*Channels*	*Dimensions*	*Senses*	*Carrier*
Cinema	1:n	Space, Time	Audiovisual	Pixels, Amplitudes
Email	1:1	Time	Visual	Letters
Web Forum	m:n	Time	Visual	Letters
Newspaper	1:n	Time	Visual	Letters, Pixels
Phone	1:1	Time	Aural	Amplitudes
Photo	1:n	Space	Visual	Pixels
Radio	1:n	Time	Aural	Amplitudes
TV	1:n	Space, Time	Audiovisual	Pixels, Amplitudes
Webpage	1:n	Space	Visual	Letters, Pixels

Table 22.1: Some Media Examples.

Table 22.1 provides a bit of motivation for the existence and increasing pop-

ularity of media theories. In particular, the twentieth century has seen a multiplication of media channels, media types and media usages. The listed types of media are distinguished by the type of information/communication channel, by the dimensions of their content, the sensual carrier and the physical carrier. Most modern media are distributed from one source to n destinations, i.e. *information media*. *Communication media* either have a 1:1 structure or an m:n structure. The latter type has become increasingly important in the digital age. Simplified we can say that most quantitative media in media understanding today are used in 1:n fashion while communication media are mostly symbolic media. Interestingly, most media are either temporal or spatial. The spatiotemporal combination is seldom – in fact, video is the only type relevant in media understanding. Of course, the sensual carrier is linked to the physical carrier.

Now, what do we want from a media theory? Very generally, we would like to understand how human beings perceive media objects, their shape, their content and their effect. All of these aspects are highly relevant for computational media understanding, since exactly these aspects have to be imitated by a smart media understanding application. Systematically, the program of media theories can be summarized in five questions.

1. How do media come into existence?
2. *What is the anatomy of a medium?*
3. *What is the content of a medium?*
4. What effects do media have?
5. How do media develop over time?

The two most interesting questions are certainly the third and the fourth. However, the first, fourth and fifth are not technical but sociological questions out of the scope of this work. In the discussion below we focus on the second and the third question as these are the most interesting for media understanding applications.

Three early birds in media theory were Walter Benjamin, Fritz Heider and Max Bense. They were mostly concerned with aesthetic aspects and consequences of – at their time – new media. In the early twentieth century, the photocopier and radio were major media innovations. Benjamin investigated the effect of copying on pieces of art. He observed that artworks (*auratic* works) loose their aura through technical reproduction – which must not necessarily be bad, since it introduces *Massenkunst*, art for the masses on an auratically lower level but, therefore, wider spread. Media understanding stands firmly on this development that was accelerated significantly by the introduction of digital media reproduction by cheap cameras in the 1990ies.

Max Bense took up the aesthetic thread and investigated whether it is possible and reasonable to define measures for the aesthetic quality of a media object. Certainly, the aesthetic quality of the Mona Lisa will be higher than of a picture in a newspaper. The operationalization of this idea leads to interestingness measures – as discussed in the second part – and similarity measurement to category prototypes. Fritz Heider went deeper into this question by asking if physical properties make something a medium, i.e. create the aura specific for a particular type of media.

Benjamin also influenced media theorists that investigated the fourth question stated above. Hans Magnus Enzensberger concluded from the auratic concept that media manipulate per se. More specifically, Neil Postman argued that media infantilize. That is, the usage of media makes the human being passive: the media acts by broadband information while the subject is reduced to a consumer, a data sink. We consider it important to note, that the methods of media understanding break up this situation by giving the subject a tool in hand that allows to take control over the consumption process, to search for specific bits of media content and to consume just this chunk instead of an entire avalanche of data.

The exponents of the Toronto school: Innis, Havelock, McLuhan and Kerckhove, developed media theory to an independent research discipline. Their ideas are based on the technical development before, during and after the second world war, the development of cybernetic theory, the formulation of the theory of dynamical systems and the invention of information theory. Harold Innis started the development by working in a similar direction as Max Bense. One of his key questions was, if media can be described in a formal/mathematical way – an idea very similar to what we do in media understanding to show that media, descriptions and categories are actually very similar concepts.

The most prominent member of the Toronto school is probably Marshall McLuhan. In his two books [260], [261] he expressed views that opened new dimensions of media theory. The following list collects a few of his famous ideas.

- The medium is the message.
- Technology changes the dimensions of space and time.
- The content of a medium is another medium.
- There is a distinction between hot and cold media.
- Hot media require little participation.
- The electrical network is a model of the central nervous system.

Not much needs to be said about the first statement. It emphasizes that the effect of a media object is not just determined by its content but also by its

anatomy, the setting and the norms of the receiver. The second statement supports what was already observed by Benjamin and others. For example, modern digital media have an accelerating effect on every day life. The third statement is very interesting for us as it supports the view that media understanding is a cyclic process. The low-level properties generate a new meaning on a semantically higher level, quantities are transformed into symbols/predicates and set into the context of the human operator.

Figure 22.2: Hot (left) and Cold (right) Media Example (© *CNBC*).

The fourth statement is of particular interest for media understanding. McLuhan distinguishes media that are *rich in details* and *poor in details*. Figure 22.2 shows two examples. The left image from the leading example, a modern newscast, is a hot medium. While the anchor person presents the news, multiple lines of scrolling text inform the viewer about international events and the development at the stock exchange. In contrast, the right image is reduced to the anchor person. In the 1960ies, a newscast could have looked like that. Rich and poor in details is a very interesting categorization for media understanding. If a medium is hot, we are able to extract a multitude of descriptions, we have polysemy, different semantic meanings and there is a potential for semantic improvement by iterative media understanding. Cold media do not have this potential. Interestingly, McLuhan named radio a typical hot medium and television a cold medium. This judgment shows the development of television over the decades. Generally, the genesis of new media goes from cold to hot (e.g. in the development of the Web). Wittgenstein antedated in the Philosophical Investigations the hot/cold differentiation. He gave the word *Freude* (German: joy) as an example for a cold medium and its reversion *eduerF* as an example for a hot medium. He concludes that the degree of novelty/surprise is an important criterion of the content. Hot media provide a constant large amount of (pseudo-)new information that pushes the receiver into passiveness. Hence McLuhan's conclusion that hot media require little participation. In fact, they allow little participation. Media understanding, as we argued above, is able to break up the dilemma of hot media (interestingness, passiveness) by destroying the constant stream and enabling interaction.

We perceive the last statement in the list of Marshall McLuhan's ideas as a symptom for the development of media theory from the late 1960ies until today: pseudo-technical argumentation. As every engineer knows, the only relevant similarity of the human brain and the electrical network is the transmission of electrical signals in a network. Neither the purpose, nor the characteristics, nor the system of transmission show significant similarities. As Sokal and Bricmont criticized in [346], postmodern constructivism lead to a misuse of physical and technical concepts in media theory and related areas. They investigate the works of Kerckhove, Virilio and others and show that most of the technically motivated statements do not make sense. Indeed, we could not extract concepts helpful for media understanding from the works of these authors. For example, Kerckhove states a fundamental relationship between alphabet and computer, which is trivial since the computer is a symbol processor and the alphabet a symbol system.

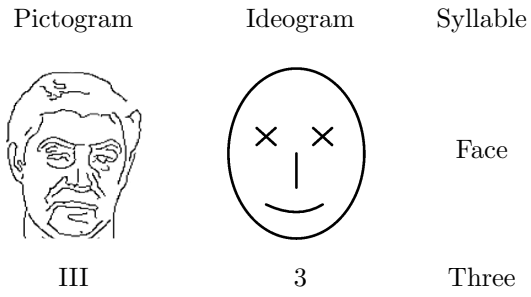


Figure 22.3: Examples for the Development of Writing.

One branch of media theory that deserves particular attention is the analysis of language and writing. Four outstanding philosophers in this area are Havelock, Derrida, Postman and Kittler. Though all four also contributed to other areas, their insights about writing systems are of highest interest to us. Havelock argued that there is no writing per se. Instead, only particular writing systems came into existence that must necessarily show at least some similarities. From his point of view, the symbol system of writing develops in three stages. Figure 22.3 gives two examples. In the first stage, pictograms are used to represent phenomena. These are then abstracted to ideograms – interestingly, a development from a hotter medium to a colder medium. Eventually, a syllable or a sequence of syllables is developed that replaces the ideogram. Depending on the writing system, the syllable may be depicted as one sign or a sequence of atomic signs.

From the epistemological point of view, the pictogram has a semantic relationship to the phenomenon, the ideogram is syntactically related and the syllable only in a pragmatic form. As the semantic relationship was criticized

as the weakest, because it is based on similarity judgment, it is only natural that a syntactical relationship is sought for replacing it. The motivation for the transformation of ideograms to syllables may be found in the standardization and reduction of the symbol system.

Havelock's ideas have inspired other authors. Derrida stated that the end of the book culture becomes visible in larger libraries. This argumentation goes in a similar direction as Benjamin's, with the book as the artwork and libraries of increasing size as a form of Massenkunst. Postman thinks into the same direction as Bense when he endeavors to measure the understandability of a sequence of symbols. Kittler points out the importance of verses for memorization. Oral cultures use verses for the transfer of information from one generation to the next. Verses are a form of redundancy. Redundancy helps learning. This supports the usage of training-based categorization techniques in media understanding as a human-like approach for better, sustainable understanding of media content.

What can we learn from media theory? Firstly, that there are periodic attempts to normalize media types and to measure their content and effect (aesthetics, for example). Concepts and similarity measurement appear to be important tools in the general understanding of media. Media theorists are still struggling for commonly accepted terms. McLuhan and others invented terms for some aspects of media before unnamed. However, the number of tools is still very small. Maybe, media understanding can contribute to media theory by providing technical insights that are valid for many types of media content. The last section of this chapter goes into this direction. In the next section, though, we introduce a small yet effective set of tools for the content-based analysis of arbitrary media objects.

22.3 Semiotics

Régis Debray, the father of mediology, defined *medium* as a combination of symbols, codes, a carrier and a memory. Carrier and memory provide the technical basis. Symbols are samples, the basic units of the medium. The *code* assigns a *meaning to symbols*. This section is dedicated to the systematic investigation of the meaning of symbols: *semiotics*. We start with introducing the elements of the semiotic system, define the possible relationships that may exist between symbols and their meaning, describe a few philosophical alternatives, give examples and, eventually, investigate a major shortcoming of the semiotic approach that will be further investigated in the subsequent chapters.

Semiotics is a research discipline with many fathers. We follow the view of Roland Barthes and Umberto Eco, i.e. we use a similar set of terms in a similar way. Our motivation is that semiotic analysis is able to investigate arbitrary media objects and to make their inner structure transparent. Thorough semiotic

analysis is a complex time-consuming task but, in particular, in the early stages of a media understanding process worth the effort. Better understanding of the recurring symbols, their meaning and interaction allows to judge the content – and, hence, the necessary methodology for analysis – more precisely than if we were ignorant of the content. A further advantage of the semiotic toolbox is that it is very small. The few general concepts allow to structure arbitrary media content in hierarchical fashion.

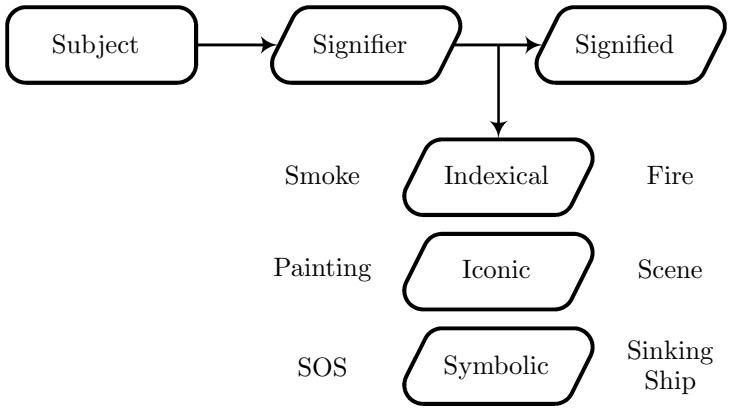


Figure 22.4: Semiotic Categories.

Figure 22.4 defines a practically usable semiotic system. In media understanding, the subject will be the experimenter or the provider of the ground truth. The subject perceives *signifiers* that represent something *signified*. The signified is the actual object/meaning. The relationship between signifier and signified is related to the one of image and phenomenon. Semiotic theory distinguishes three forms of relationships (codes).

- *Indexical*: The signifier allows a *causal* conclusion on the signified. In the example, the presence of smoke indicates a fire.
- *Iconic*: The signifier and the signified are somehow similar, for example, like a classic oil painting and the pained scene.
- *Symbolic*: The relationship between signifier and signified is arbitrary but well-defined. A typical example is the international distress signal SOS which indicates someone in the need for help.

The image theory distinguishes syntactic, semantic and pragmatic relationships between image and phenomenon. Though these two views are very similar, they are not the same. Not every accepted signifier is an image and – due to

polysemy, for example, not every image is an appropriate signifier. We see the similarity of the two types of relationships as follows.

1. An iconic relationship between signifier and signified may indicate a semantic and/or syntactic relationship between image (signifier) and phenomenon (signified). In Chapter 28 we will come across dual process models (already briefly mentioned in the second part) that implement this idea and operationalize iconic relationships as combinations of semantic and syntactic properties.

2. The pragmatic relationship between image and phenomenon may indicate an indexical and/or symbolic relationship between signifier and signified. All three types of relationships are causal. The equivalence of pragmatic and symbolic appears more natural, but indexical cause-effect relationships also belong to this group.

The practical use of the two terms (signifier, signified) and their three relationships is straightforward. Being conscious of the differences between the two terms allows us in the first step to recognize objects, for example, in film as signifiers for some meaning. In the second step, we identify the meaning as the signified. Eventually, in the third step the relationship between signifier and signified is chosen as the best fitting from the list of options. Table 22.2 gives a few examples of signifiers, signifieds and relationships for the media types considered in this book.

Medium	*Signifier*	*Signified*	*Relation*
Audio	Siren	Danger	Symbolic
Bioinfo	Start Codon	Begin of Gene	Indexical
Biosignal	ECG Pattern	Healthy Heart	Iconic
Image	Circle	Ball	Iconic
Stock	Sinking Value	Near Crash	Indexical
Text	(sic!)	Last Word Correct	Symbolic
Video	Face of a Clown	Funny Movie	Indexical

Table 22.2: Examples for Semiotic Analysis.

The majority of the examples in the table suffer from one major deficiency: polysemy. In most cases – in particular, if the relationship is not indexical – two or more meanings could be assigned to a signifier. For example, a siren may indicate an approaching ambulance car or a change of shifts. The true meaning depends on the context. Similarly, a circle may stand for the sun, the moon, a pill, a disk, etc. The face of a clown may indicate a comedy (if it is a nice clown) as well as a horror movie (if it is a nasty one). The context is decisive.

The context of a signifier is not always known. If not, we can at least give probabilities for the most likely meaning and the less likely ones. An established system here is the distinction in *denotation* and *connotation*. The denotation of a signifier is mostly influenced by direct experience and the cultural imprint. Within a sign system, denotations are rather stable and well established. Connotations may be seen as add-ons that may amplify, weaken or even reverse the denotation. For correct semiotic analysis it is important to understand that polysemy exists in signifiers. Hence, where possible the analyzer should make the exact meaning of the signifier explicit by selecting the appropriate categories from the denotation and all connotations of the signifier.

The relationships of image and phenomenon and the relationships between signifier and signified are two ways of looking at the gap between objective reality and subjective experience. We would like to introduce two more options that were developed by philosophers. Aristotle defined causality very widely – in contrast to *indexical* above – by allowing four possible relationships between the idea and the reality of something.

- *Causa formalis*: The idea describes the structure of the reality. Hence, causa formalis is equivalent to the semantic/iconic relationship.
- *Causa finalis*: The idea indicates the use of the reality, which makes it equivalent to the pragmatic/symbolic relationship.
- *Causa materialis*: The idea describes the building blocks of the reality as a syntactic/iconic relationship.
- *Causa efficiens*: The idea indicates the consequences of the reality, i.e. it is indexical.

Wittgenstein supposed another, simpler system in the form of *family resemblances*. Targeted at the common properties of related stimuli, family resemblances are a form of semantic/syntactic (i.e. iconic) relationship. We see them as a form of similarity measurement by dual process models. See the chapter on human similarity perception for details on this idea. Wittgenstein did not consider indexical, symbolic or pragmatic relationships.

Now, what is the use of all these ideas for media understanding? We named already the benefits of using semantic analysis as an entry point for solving a media understanding problem. Furthermore, semiotics can be used to analyze descriptions, class labels and ground truth. Very generally, descriptions and class labels can be seen as signifiers for media objects and ground truth judgments. The semiotic analysis of these ingredients of the media understanding process allows to identify polysemy as one trap and semantic insufficiencies in the data as another.

Figure 22.5: Thematic Taxonomic Bridge.

We have seen above, that iconic semiotic relationships cover both syntactic and semantic image relationships. There is a danger in this insufficiently precise definition. Figure 22.5 gives an example for the problem. Ball and balloon are related in an iconic fashion, in particular, semantically. Psychologists call this relationship *thematic*. Balloon and airplane are also related iconically/syntactically, which is called *taxonomic* in psychology. Hence, we have a chain of two iconic relationships. However, ball and airplane do not have an iconic relationship. At most, we can think of a symbolic one. This problem is called *thematic taxonomic bridge*. It is very important in similarity measurement. Here, it indicates that despite all efforts the terms used in semiotic analysis are not sufficiently precise and, therefore, have to be used with caution.

Semiotic analysis has its origin in text understanding. However, the method is applicable on all types of media. It allows to analyze the content and structure of arbitrary media objects which makes it a valuable tool for the early stages of media understanding. In particular, it helps to understand hidden polysemy in the media content. In the last section of the chapter, we move away from qualitative analysis and endeavor to formalize the usage of media analysis with the help of information theory.

22.4 Media and Information

This section is a first attempt to merge the results of media theory and semiotic analysis with those of information theory. Media theory and semiotics aim at *extracting symbols and their meanings from media objects*. Information theory estimates the interestingness of data streams. The relationship of these two disciplines is of *pragmatic* form, i.e. information theory makes use of the results of the other disciplines. The objects named and described by media theoretic and semiotic methods can be used as the units/events investigated by information theory. In this section, we would like to make this idea transparent. We proceed in the following steps. First, we summarize the types of relationships that may exist between the perception and the reality of something. Then, we describe information-theoretic developments in media theory and semiotics. Eventually, we merge the two areas in a flow model for sign-based media understanding.

In the first three sections, we encountered a number of differentiations between the perception (image, signifier, cause) and the reality (phenomenon, sig-

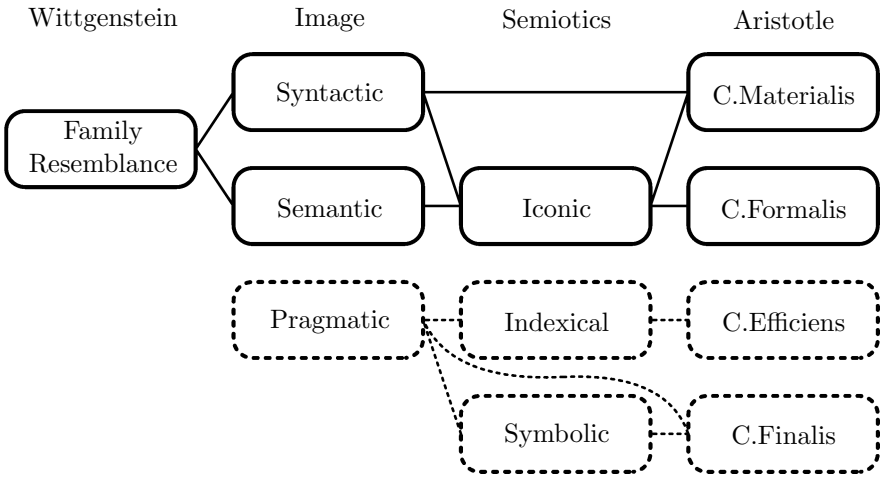


Figure 22.6: Relationships of Perception and Reality.

nified, effect) of some entity/event. Figure 22.6 summarizes and connects the possible relationships as argued in the preceding sections. Those types that we consider most relevant for media understanding are printed with full borders, the others with dotted borders. The figure shows that all approaches except Wittgenstein's go beyond similarity-based relationships. However, since those are the ones that are most relevant for media understanding – eventually, we want to recognize perceivable pattern similarities – they have to be considered with special attention in semiotic analysis in media understanding.

Pattern Type	*Low-Level Examples*	*High-Level Examples*
Point	Average, Maximum, Peak, Determinant, Gradient	Interest Point, Location, Symbol
Interval	Histogram, Deviation, Density, Range	Template, Contour, Phrase, Gene
Group	Skew, Regularity, Correlation	Symmetry, Sentence, Self-Similarity

Table 22.3: Some Signifiers in Media Understanding.

Semiotic analysis of media understanding categories means working on a semantic level that is significantly lower than in the normal case. Table 22.3 provides a list of typical descriptions (signifiers) used in media understanding. Obviously, the simpler the type of signifier, the higher the potential for mis-

understanding (polysemy). Hence, the interpretation depends heavily on the context of the description. More complex aggregates (e.g. groups vs. points) decrease the risk of misunderstanding. So do high-level signifiers (e.g. interest points vs. peaks). In consequence, semiotic analysis of such signifiers will be more successful if the descriptions are of a more complex type on a semantically higher level. The actual analysis will, of course, analyze the type of relationship between signifier (the description) and signified (the object to capture) and, most importantly, the degree to which this representation is achieved. In order to be useful in the world of media understanding, this analysis should be as formal as possible.

So far for motivation. The development of media theory and of semiotic theory has a branch of (quasi-)information-theoretic investigation/thinking. This direction is mostly represented by the works of Vilém Flusser. His ideas are influenced by the developments in cybernetics, dynamical systems and information theory between the wars and after the second world war. Before we investigate the ideas of Flusser, we would like to describe the environment briefly.

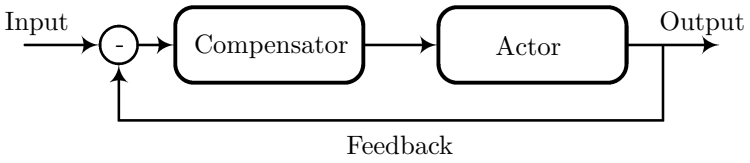


Figure 22.7: Feedback Loop.

Cybernetics is based on the feedback system. Figure 22.7 gives an example. This non-linear system uses the output of the actor to manipulate the input signal in the compensator. Simple feedback systems are, for example, echo machines. Norbert Wiener applied the theory of feedback systems on other scenarios, including the behavior of animals and humans. The development of the mathematical theory of dynamical systems is closely linked to cybernetics, since dynamical behavior is often a consequence of a system with a multitude of (sequences of) feedback channels. The statistic/mathematic treatment of such systems allows for the description of their overall behavior (e.g. the formulation of attractors, see Chapter 27). Such ideas were a substantial inspiration for media theory.

Another inspiration comes from ergodic theory, i.e. the theory of working systems. In simple words, an ergodic system is one (expressed by a set, an algebra, a flow function over time and a measure) that requires all non-trivial subsets of the set to change over time. Ergodic theory has been used to describe oscillating systems. This path leads directly to the information theory of Shannon, who understands the signal-generating source to be an ergodic system.

On this foundation, media-theoretic considerations were formulated by several authors. McLuhan's distinction of hot and cold media is actually an information-theoretic idea. It defines a fundamental scale for the interestingness of media content. Applied to the technical signifiers listed above, we can, for example, characterize point descriptions as cold. Groups, rhythms, etc. will be significantly hotter. Descriptions on a semantically higher level are hotter than elementary descriptions, and so on.

Flusser went further into the information-theoretic direction. Explicitly referring to the technical definitions, he made, for example, the following statements.

- Information is the emergence of the improbable. Information means to carve shape into something.
- Low entropy is order, continuity, availability, uncertainty, information.
- There is a negentropy of all life. Only the human is able to invert entropy.

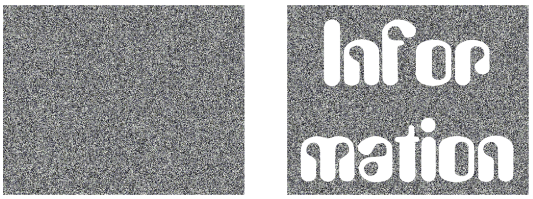


Figure 22.8: Information is Improbable.

The first statement sees information as the result of an ordering process. Figure 22.8 illustrates this idea. The left image shows gray noise. The right one has the same entropy on the pixel level but it clearly shows some text information. The text has been carved into the gray sand. Luhmann expressed the same idea on the dialectic level when he said that communication is improbable.

The second statement from the list is of central interest for media understanding. Low entropy will certainly mean a high degree of order and information. The Shannon definition punishes every deviation from uniform distribution of all elements of the set of symbols. Information is such a deviation. Availability and continuity can be seen as temporal aspects of this equivalence. However, it is not clear why uncertainty should match with low entropy. Generally, something with higher availability and durability should be less uncertain. We believe that *uncertainty* here is used as a synonym for *improbable*, though the meaning is in this context clearly different.

The last statement expresses the implication of the laws of thermodynamics that if everything (every process, life) strives for disorder we have not reached

this point yet and, hence, a certain potential for negentropy must still exist. The second sentence remains obscure. As we know, negentropy is not inverted entropy but a contrast from the theoretical maximum to the actual value. Optimization of entropy is certainly not a particularly human ability – rather the opposite. The examples show that Flusser's idea provide an interesting bridge from media theory to information theory.

How can we make use of these thoughts for media understanding practice? As already sketched, we suggest a process with three steps.

1. Semiotic analysis of the media content under consideration with media-theoretic concepts (hot/cold, information/order, etc.) in mind. Analysis will include the recognition of signifiers from media events and the definition of the relationships between signifiers and signifieds. The result of this step is a list of objects that can be used as units in the second step.

2. Information-theoretic analysis of the media content. This includes the computation of probabilities of occurrence for semantically relevant signifiers and the computation of (conditional) entropy values as interestingness measures. The result of this step is sensitivity for the magnitude of the media understanding problem.

3. Media understanding operationalization of the acquired results. How can the signifiers be modeled by feature transformations? How can the relationships be modeled by categorization processes? How can the interestingness values be employed for evaluation? The answers lie in the application of the introduced media understanding tools with sensibility for the complexity of the understanding problem.

In conclusion, media theory, semiotic analysis and related areas of research evolve in a climate substantially different from media understanding. Nevertheless, it is worth the effort to identify and use major results of these disciplines for pushing the frontiers of media understanding. The most important result of this chapter should be a better understanding of the complexity of media objects and their content. Semantics and context are expressed on various levels in different forms. Successful media understanding has to deal with this complexity.

Chapter 23

Perception and Psychophysics

Lists fundamental aspects of human perception, shows where perceptual and cognitive insufficiencies of the human brain lie, gives an introduction into the psychophysical model and discusses psychophysical aspects of hearing and vision.

23.1 Human Perception and Cognition

In the last chapter, we approached the phenomenon *media* from the philosophical direction. Cognitive aspects were not considered. This is the task of this chapter (and of Chapter 29 – for low-level neural processes). Our motivation is straightforward. Human perception and cognition are not objective in the sense of physical processes. We do not lay the same weight on all stimuli. Rather, we focus on particular aspects of perception, emphasize some stimuli and neglect others. It is paramount for media understanding to take the peculiarities of human perception into account. In the first and second part, we already introduced a number of concepts that cannot be ignored in feature extraction and categorization. In this chapter, we fill the gaps and present a sketch of perception and cognition that should include everything required for professional media understanding.

As always, the chapter is organized in four sections. The first provides an overview over perception. The second section deals with particular problems of perception and cognition. We distinguish three types of errors that are introduced by examples, explained and discussed. Section 23.3 gives a systematic

introduction into psychophysics, the expression and explanation of psychological phenomena of perception by physical laws. The last section specifies and investigates the psychophysics of the two senses most relevant for media understanding: hearing and seeing. Psychoacoustics and the psychology of vision are two important frontiers of psychophysical research. Many important results of these disciplines have already found their way into media understanding. We are positive that further significant advances will be made possible through understanding this frontier of media understanding better. The present chapter is dedicated to contribute to this end.

Below, we discuss the fundamentals of the auditory sense, vision and the other senses – which are due to insufficient hardware, unfortunately, still out of the scope of media understanding. The purpose of this section is to gather the relevant bits of information in one place. We do not intend to provide a full introduction into the anatomy of human senses. Many excellent descriptions can be found in the literature. In contrast, our focus is on understanding the signal processing procedures applied in the individual senses.

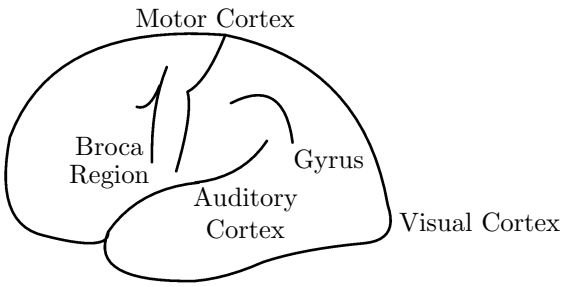


Figure 23.1: The Human Brain.

Every discussion of human perception requires a basic understanding of the human brain. For a start, Figure 23.1 sketches its anatomy and names a few centers of perception. Details on neural processes can be found in Chapter 29. Generally, the brain is divided into the *autonomic nervous system* and the *somatic nervous system*. While the first is responsible for the basic management of life (e.g. the control of the heartbeat), the second includes all perceptive and cognitive processes. The brain consists of two *hemispheres* which are connected by the *corpus callosum*. The corpus callosum is, for example, very important for visual perception, as we will see below. Altogether, the brain consists of 10^{10} neurons which are connected by approximately 10^{13} *synapses* and *dendrites*. Considering the size of the human brain, the average density is 10^5 neurons per cubic centimeter. The average neuron is connected to 3% of its neighbors in a sphere with a diameter of 1mm.

Sense	*Number of Receptor Cells*
Aural	3.10^4
Gustative	8.10^5
Haptic	2.10^6
Olfactory	4.10^7
Visual	2.10^6

Table 23.1: Receptor Cells per Sense.

Table 23.1 lists the number of receptor cells per sense. Astonishingly, the arguably most effective sense, hearing, requires the smallest number of cells. In contrast, the olfactory sense – which is often perceived as ineffective and outdated – requires by far the largest number of receptor cells. The gustative, haptic and visual senses lie somewhere between these extremes. We can only speculate about the reasons for these facts. As we have already seen in the discussion of the cochlea organ, the auditory sense is very efficiently built. It implements a procedure very similar to the Fourier transform. The hair cells and tip links are a simple analog to digital converter that separates critical frequency bands. Such mechanisms are hard – if not impossible – to implement for the other senses. The visual scanner, for example, has to aggregate simultaneous stimuli, where the auditory sense only has to resolve the components of one stimulus. That is, the low dimensionality of the sensual carrier of auditory stimuli may be seen as one reason for the low number of receptor cells required.

Little is known of the development of the human brain. It has been observed that the growth – in terms of the number of neurons – is exponential between birth and an age of 2.5 years. After this period the growth in neurons decreases until the age of 12 years. In parallel and after this period, the growth rate of the number of neural connections remains more or less constant, which indicates that human learning is a question of neural connections, not of the number of neurons or age.

The brain is an analog to digital converter that converts stimuli into *action potentials*. Loaded neurons show activity for approximately 2ms and a potential of 50-80mV. The speed of event propagation is delayed to circa ten meters per second by the neurotransmitters that implement the connections. The weight of an impulse depends on its distance from the cell center (axon hill) and ranges between one and five percept of the action potential. The total *perceptual load* produced by all receptor cells of all senses sums up to approximately *1GB per second*. Being able to process this avalanche and remaining sane requires our cognition to throw away the largest part of these data.

We have already discussed various aspects of the auditory sense. In the first

part of the book, we introduced three fundamental psychoacoustic properties: the absolute thresholds of hearing, the perceived loudness in contrast to the sound pressure level (sone curve) and the perceived pitch in contrast to give sound frequency (mel curve). In the second part, we explained the conversion of auditory waves into electrical nervous signals by the cochlea organ, and we introduced the bark scale for the critical bands of hearing.

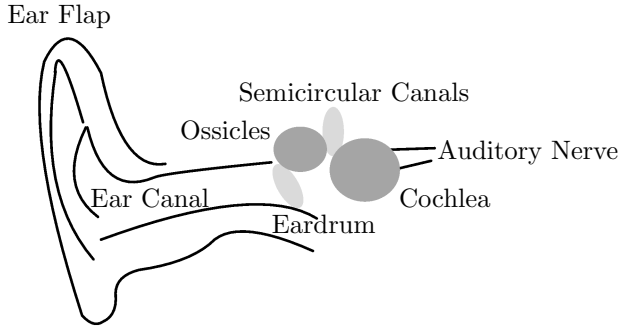


Figure 23.2: Human Hearing System.

These advanced concepts of hearing that are highly relevant for audio compression and audio feature extraction are mostly consequences of the anatomy of the ear. Figure 23.2 provides a sketch. The ear is usually split into three sections: outer, middle and inner ear. The outer ear consists of ear flap and ear canal. Its anatomy acts like a weighting function that emphasizes some frequency components while suppressing others. The middle ear consists of eardrum and the ossicles. It provides the transformation of the auditory air wave into the fluid wave of the cochlea. The inner ear spans from the cochlea organ to the auditory nerve and works as described in the second part.

Most relevant aspects of the early stages of hearing have already been discussed. Hearing is optimal at around 4kHz. The sound pressure level (SPL) is computed relatively to a reference sound of 1kHz.

$$SPL = 20. \log \frac{p_{sound}}{p_{reference}} \tag{23.1}$$

The main cognitive processes happen in the auditory cortex. It is well studied that – maybe because of their efficient representation – large parts of heard music are stored for recognition in the cortex. Two further regions connected to the ear are the gyrus and the Broca region. The first, together with the semicircular canals, is responsible for the human sense of balance. The second region, in combination with the so-called Wernicke region, is responsible for speech processing and language understanding. Unfortunately, only little is yet

known about the details of the audio understanding process implemented in the human brain.

Like for audio, several aspects of human vision have already been described in the first two parts of the book. For example, we introduced the three stimuli theory for color representation and saccadic seeing as a form of scanning. We explained the different wavelengths on which the three types of cones respond, their organization in the fovea, various color models including the RGB scheme that is most similar to the cones and, eventually, the early processing of edges in the visual system. These facts are enriched in the subsequent paragraphs.

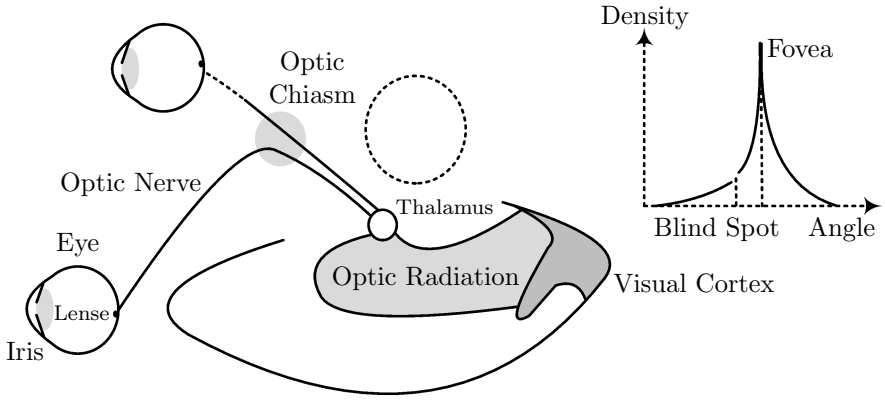


Figure 23.3: Human Visual System.

Figure 23.3 illustrates the visual system. The *muscles of orbit* control the orientation of the eye and the size of the lens. The iris is an aperture that controls the amount of light that reaches the eye background (*retina*). In the retina, light beams with wavelengths between 400nm and 720nm are converted into electrical signals in a three layer process. Rods and cones respond to brightness and particular wavelengths, respectively. Bipolar cells react on the absence/presence of light. Ganglia cells sum up circular neighborhoods of bipolar cells to on/off structures. The density of the retina is not uniformly distributed. The right diagram of Figure 23.3 shows that, depending on the angle, the density is maximal in the fovea (mostly cones) while it is relatively low outside (mostly rods). The fovea covers only 1% of the retina with 5.10^4 ganglia cells per square millimeter. Outside the fovea, the retina has a density of 1000 cells/mm^2. There is a blind spot where the optical nerve exits from the eye.

Stereo vision requires the existence of two signals with spatial disparity. The signals generated by the two eyes are propagated to the corresponding thalamus but – through the optic chiasm – also to the opposite one. In the back of the brain lies the visual cortex where the later stages of vision are located. It is

interesting to note that a field of optic radiation lies between thalamus and visual cortex that is influenced by the optical signals.

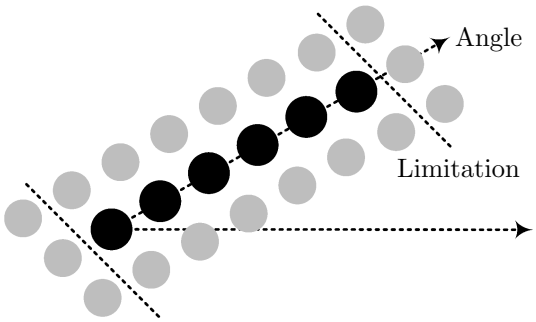


Figure 23.4: Primary Visual Cortex Pattern Example.

Figure 23.4 illustrate the result of the first three stages of visual perception. The reaction frequencies of the three types of cones were already discussed in the first part. The differentiation between red and green cones took place only 35 million year ago (by mutation) which is the reason why the reaction wavelengths of these types of cones are so close to each other. The signal from the receptor cells is propagated to two types of bipolar cells. *On bipolar cells* fire if a light stimulus is present, *off bipolar cells* fire otherwise. That is, the latter type of cells generates a signal where there is no outside stimulus. The input of these bipolar cells is fed into accumulating ganglia cells that have a structure very similar to an interest point operator (e.g. the Laplace operator). There are two types of ganglia cells. *On ganglia cells* define an *on center* in a circular *off neighborhood* while *off ganglia cells* implement the inverse pattern. Aggregating these patterns in later stages allows for the recognition of edges, which are distinguished by their quantized angle and their length (ends within focus/otherwise). The existence of this pattern supports the usage of interest points as the first step of object detection in media understanding.

What comes in later stages of visual perception is hardly known. It has been found out that particular types of ganglia cells are sensitive to different color tuples (e.g. red/green, blue/yellow), some cells only fire only if the optical signal represents a face, hand or another part of the human body. Furthermore, the output of the ganglia cells is not the only input of the visual cortex. In the temporal lobe, the optical signal is also processed directly. The results are firing patterns particular for certain groups of objects and events. However, it has to be stressed that visual objects are never represented by a single neuron ('grandmother neuron') but always by firing patterns (*spike trains*, see Chapter 29).

In conclusion, the system of human perception and cognition is complex and

so far only the earliest stages of processing are well understood. The sensory load is exceptionally high – even though the human eye, for example, is a sequential scanner and not a parallel one – as in case of the fly. There is not much to say about the senses of smell, touch and taste. The majority of touch cells are located on the lips and the hands. Separate regions are responsible for basic tastes: salty, sweet, sour, bitter and some particular acids. Unfortunately, we do not have appropriate sensors for the detection of these dimension of reality by computer systems. Hence, they are only of minor interest for media understanding.

23.2 Perceptual and Cognitive Errors

Human perception and cognition are complex – and not always right. This section serves as the transition from the physiological description of perception and cognition in the last section to the psychological description in the next section. We point out that three types of errors appear in human perception in various forms.

- *Perceptual-physiological* illusions
- *Cognitive-perceptual* illusions
- *Cognitive-statistical* illusions

The complexity of these illusions increases from level to level. While most perceptual-physiological illusions can easily be explained by deficiencies in the early parts of reception, we have only a limited understanding of cognitive illusions of both types. Below, we introduce several examples for all three types of perceptive errors. The discussion should make clear that our perception does not work as precisely as the physiological description of the receptor organs suggests. Some errors can be considered shortcomings of human perception. Others are rather features of perception. In particular, the third type of error has to be taken into account in media understanding in order to imitate human perception and judgment correctly. *Man is the measure* in media understanding. A *supersemantic* media understanding application (better than human perception) will produce as little user satisfaction as a *subsemantic* one (worse human perception/cognition). See Chapter 25 for a discussion of these ideas.

The simplest type of perceptual errors can often be explained by the particularities of the early stages of perception, in particular visual perception. Figures 23.5 and 23.6 show two well-known visual perceptual-physiological illusions. Watching the Hermann grid without focussing on a particular point should create the illusion of gray dots in the intersection points of the white lines. The dots should vanish when focussing on them. Until recently, it was

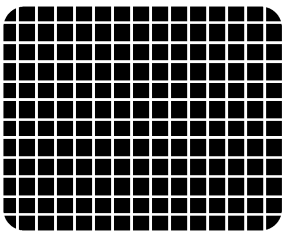


Figure 23.5: Hermann Grid Illusion.

commonly accepted that *lateral inhibition* causes this effect. Lateral inhibition is a – often, useful – neural function, in which the firing neuron with the highest potential inhibits its neighbors, thus enabling the precise localization of the stimulus. Excitatory and inhibitory ganglion cells implement lateral inhibition. According to this approach, the gray dot is the result of central inhibition in an off ganglion cell. However, recent experiments with waved lines suggest that this cannot be the reason for the Hermann grid illusion. Instead, the explanation is sought in the visual cortex.

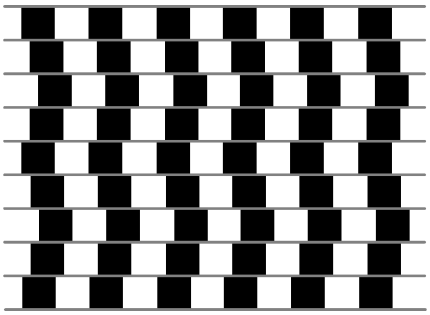


Figure 23.6: Café Wall Illusion.

Figure 23.6 shows the famous café wall illusion. Watching the figure without focussing should create the impression of non-parallel gray lines. This illusion can satisfactorily be explained by *border locking*. Visual perception seems to rely on object contours for stabilizing the perception of objects in saccadic seeing and during human motion. Border locking appears to fail, if the contrast between objects is too low or too high. The gray lines between the black and white tiles cause both of these problems.

Is it advisable to imitate these illusions in visual media understanding? Probably, not. Perceptual-physiological illusions do not express relevant aspects of human perception which is supported by the fact that almost all of them can be

removed by focussing on the stimulus the causes the illusion.

Figure 23.7: Gray Bar Illusion.

Figures 23.7, 23.8 and 23.9 illustrate several well-known cognitive-perceptual illusions. The first is the gray bar illusion. The bar in the center appears to be a brightness gradient while it really is of unicolor. The reason is a cognitive function that creates color and brightness constancy under varied lighting conditions by compensation. The outer bar suggest the existence of a light source at the right of the figure. If the gray bar in the center is under these conditions unicolor than it must really be a gradient with the darkest part closest to the light source.

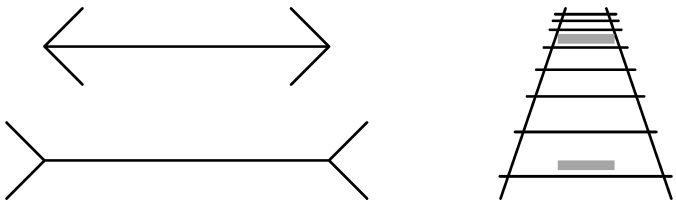


Figure 23.8: Depth Illusions: Müller-Lyer (left) and Ponzo Illusion (right).

The two depth illusions in Figure 23.8 are also results of human cognition. The left Müller-Lyer illusion suggests that of the two horizontal lines the upper one should be shorter while, in fact, they are of exactly the same length. The reason for this illusion are the angles that are interpreted by human cognition as object contours. Hence, we perceive the upper figure as emersed, the lower as the opposite. If under these conditions the edges are of equal length, the lower one must in fact be longer since it is further in the back.

Then Ponzo illusion implements the same idea. Of the two gray bars the upper one should be perceived longer. The reason is depth perception induced by the rails. The upper bar is closer to the vanishing point, i.e. further to the back. Since it has equal length in the figure, it must in reality be longer.

Eventually, Figure 23.9 shows a third group of cognitive-perceptual illusions: illusory figures. The triangle (left) and the square (right) are in fact not there. Their contour is only suggested by the contours of the displayed objects. The exact reasons for the perception of illusory figures are yet unknown. We believe,

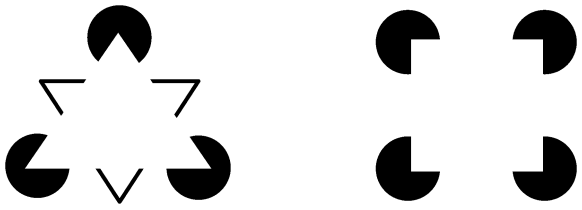


Figure 23.9: Illusory Kanizsa Figures.

that the mechanisms responsible for Gestalt perception are also responsible for illusory figures. Groups of aligned edge segments are here joined to figures – which is an application of the grouping laws. The existence of illusory figures supports our argumentation in the second part that interest points should not only be selected by high curvature but also by Gestalt laws. By that we would become able to recognize partially hidden objects such as the presented illusory figures.

We conclude that it makes sense to imitate human perception of cognitive-perceptual illusions. Depth perception, color contingency and Gestalt laws are relevant aspects of human dealing with reality. These aspects should be made part of media understanding systems.

The last group of cognitive errors that need to be discussed here are not directly connected to perception, rather results of experience and aggregation. The cognitive-statistical illusions have been identified in psychological research on human choice behavior, human similarity perception and the development and application of norms. Interesting works include [182] and [183].

Choice, similarity judgment and related problems are usually based on cognitive references, so-called *norms*. Norms are mixtures of probabilities like mixture models. One norm defines one quantitative category, e.g. the typicality of some item for a class of objects. Humans build norms for all events that are relevant in their decision space and that appear with a certain frequency. In media understanding terms, norms are comparable to the categorization model that is used to conclude from descriptions on semantic categories.

Unfortunately, the development and application of norms is not strictly rational. Psychological experiments have revealed surprising insufficiencies in human decision making and similarity judgment. The problems can be grouped into three clusters.

- Problems of *representativeness*
- Problems of *availability*
- *Anchoring* problems

Problems of representativeness are closely connected to similarity measurement, because, usually, the dependence of one stimulus on another (e.g. co-occurrence) is determined by the representativeness of the latter for the first, which is operationalized as similarity (e.g. in the human choice model, see Chapter 17). Problems of representativeness include human insensitivity to *a priories*, human belief in predictability, sensitivity to worthless evidence, misconceptions of chance and ignorance against regression towards the mean. These aspects are explained in the next paragraphs.

Human insensitivity to a priories refers to the fact that we usually do not take the base probabilities of events into account. For example, in the first part in Chapter 9 we introduced the HIV test example. Bayes theorem reveals that even in case of a positive test it is relatively unlikely that the tested person is HIV positive (in a first world country). The reason is the *a priori*: It is generally very unlikely that someone is HIV positive because so few people are. This a priori is usually not considered by humans.

Human belief in predictability and *misconceptions of chance* are two related insufficiencies. Both problems refer to the fact that humans tend to infer conclusions from insufficient data. For example, the belief in a 'law of small numbers' in analogy to the law of great numbers, is nonsense. In the casino, after a series of ten times red, it is not at all more likely that black will appear more frequently in the next rounds. This is a misconception of chance. Human belief in predictability is to conclude on something from such inappropriate ground.

Furthermore, humans are prone to be influenced in their choice behavior by *worthless information*. The fact that a particular sports star advertises a car is no reason to buy this car. Eventually, *regression towards the mean* is a well-studied law of nature that appears in many different contexts. Humans tend to ignore this law. For example, if both parents are taller than average it is not likely that their child will be even bigger. It will rather be smaller than the parents.

The availability problems refer to the norm itself. Is it available and trustworthy? Problems include the so-called judgmental heuristic of validity, sensitivity to prominent examples and search set complexity. The *judgmental heuristic of validity* refers to the problem that humans tend to estimate the likeliness of some event by the availability of examples. In short, we see what we know: our paradigms. The *sensitivity to prominent examples* is closely linked to the *typicality problem* (concept theories). We tend to consider a famous sports star to be the typical athlete while, in fact, the typical athlete is rather an average jogger. *Search set complexity* describes the problem that we can easily give references for some norms but hardly for others. For example: How many words do you know that start with the letter 'r' and how many that have this letter at the third position?

Eventually, anchoring refers to the phenomenon that the results of human reasoning often depend on the starting point. For example, test subjects gave significantly different estimates for the number of countries in the United Nations if a first hint was chosen very low or very high. Furthermore, experiments could show that humans tend to give higher likeliness to joint events. Obviously, an event that joins two probable events can never be more likely than one component. However – maybe as a result of the worthless evidence problem –, humans build such norms.

It is a difficult to answer question whether or not media understanding should try to imitate human cognitive-statistical illusions? Some aspects are certainly just errors (e.g. sensitivity to worthless evidence) while others may be seen as valuable particularities of human reasoning (e.g. the sensitivity to prominent examples and the anchoring problem in general). Sophisticated similarity measurement techniques already take certain aspects into account. For example, the insensitivity to *a priories* is eliminated by the application of Bayesian networks and Markov processes. We believe that this question is a true frontier of media understanding, in particular for the abstract statistical data types such as stock data and bioinformation. The identification of the best mix of intentional errors and supersemantic behavior is a worthwhile research undertaking.

In summary, several deficiencies exist on the levels of human perception and cognition. Some of them are worth consideration in media understanding while others can be ignored/corrected without any loss of authenticity. In the next section, we go one step further than just describing individual phenomena of cognition and introduce a general theory of psychological perception.

23.3 Psychophysical Theory

Psychophysics is a research discipline that developed out of physical research (in particular, optics) in the early nineteenth century. The idea was to investigate human perception of physical stimuli and to describe this perception in as few and as simple as possible laws. In the preceding sections we investigated the anatomy of the human reception apparatus and examples for its failure. Psychophysics is there for the structured representation of this knowledge. Below, we first introduce the psychophysical model, list the major research questions and explain the major findings: the laws of psychophysics.

Figure 23.10 illustrates the *psychophysical model*. Some outside stimulation ϕ is received and represented by human perception ψ. This process is called *outer psychophysics*. *Inner psychophysics* refers to the cognitive processes applied on the perceived stimuli: the neuroprocess. The laws of psychophysics refer mostly to outer psychophysics. Inner psychophysics is today a part of neuroscience.

The fundamental hypothesis of psychophysics is that $\phi \neq \psi$, i.e. we do

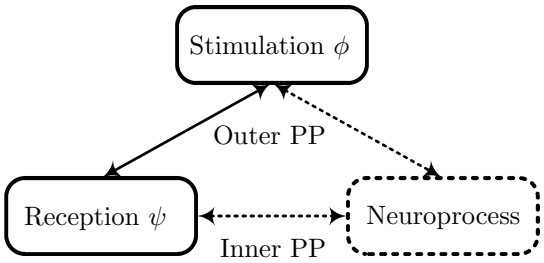


Figure 23.10: Psychophysical Model.

not perceive reality as it is. The examples given in the last section support this hypothesis. However, psychophysical research goes one step further and formulates laws that answer the following research questions.

1. What are the *absolute threshold*s of human perception?
2. What are the *thresholds of discrimination* of human perception?
3. What are the *just noticeable differences* of human perception?

The third problem is also referred to as *scaling*. Below the just noticeable difference (JND) lies a space that is limited by the *point of subjective equality*.

Absolute thresholds and JND could so far hardly be described by compact mathematical laws. Instead, they have been documented per sense and for the full set of search space parameters. We already mentioned the absolute thresholds of hearing that depend on the frequency of the sound. Similar investigations have been conducted for brightness perception, color perception, the perception of smell, taste, etc. In media understanding, these data are highly valuable for the calibration of capturing devices and the preprocessing of media content for feature transformation.

For the expression of discrimination thresholds, Weber defined the following law (referred to as the *extended Weber law*).

$$\Delta\phi = a_1.\phi + a_2 \tag{23.2}$$

That is, the size of the stimulus determines the size of the difference required for recognizing the change. The bigger the stimulus, the higher the required discrimination threshold. The parameters a_1, a_2 are constants for a particular sense and application. The Weber law works astonishingly well for as different applications as audio perception and the perception of sweetness.

The Weber law does not yet provide a transformation from the outside world to the perceived world. The *Fechner law* defined this transition for the first time.

$$\psi = a_1 . \log \phi + a_2 \tag{23.3}$$

The meaning of the parameters is the same as for the Weber law. The perception of stimuli is generally claimed to be the logarithm of the physical stimulus. That is, there is a general absorption of stimulus components in the reception process. The Fechner law works well for some types of stimuli, but fails for others. Therefore, the Fechner law has been replaced by Stephen's power law.

$$\psi = a_1 . \phi^{a_2} \tag{23.4}$$

The exponent $a_2 \in [0, \infty[$ is referred to as Stephen's exponent. Stephen's power law allows a vide variety of different behaviors. It includes the inhibitory Fechner law for $a_2 < 1$, linear/direct transformation for $a_2 = 1$ and excitatory stimuli for $a_2 > 1$. Stephen's exponent has been identified for many types of stimuli. A complete overview can be found in [142] on page 303. Figure 23.11 shows a few interesting examples.

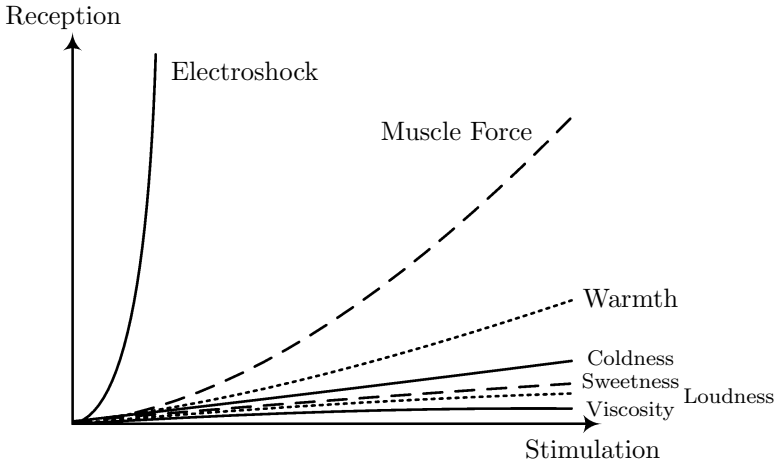


Figure 23.11: Examples for Stephen's Exponent.

The feeling that cold metal creates on the skin has an exponent $a_2 = 1$. That is, perception ψ scales linearly with physical reality ϕ. In comparison, the perception of warm metal has an exponent $a_2 = 1.6$. That is, warm metal creates a stronger impression than cold metal. Practically, warm things become earlier unpleasant than cold things. This fact may be explained by higher tolerance against coldness required from mammals by nature and evolution.

Muscle force and electroshock are two examples for excitatory stimuli. In both cases a small increase in the stimulus is sufficient to create an over-linear re-

sponse. Electroshocks almost immediately become unpleasant, weights soon become too heavy – even for the best-trained athlete. We conclude that a Stephen's exponent above unit size stands for small tolerance against such stimuli.

On the opposite end of the scale we have the perception of sweetness, loudness and viscosity. All three stimuli are perceived under-linearly. That means for sweetness that a small increase is hardly perceived. Larger increases result in smaller sensations. In order to perceive a linear increase in loudness, the sound pressure level has to be increased over-linearly. That is, the sone curve is an application of Stephen's power law for $a_2 = 0.67$. Viscosity perceived on the skin has one of the lowest exponents: $a_2 = 0.42$. That is, even a large increase in viscosity is hardly perceived any more.

Stephen's power law appears to be a satisfactory solution for the description of discrimination thresholds. Together with tables and charts for absolute thresholds and just noticeable differences, psychophysics provides valuable information for media understanding. In particular, feature transformation in audio and visual media understanding benefits significantly from the introduction of psychophysical knowledge. Some bits are well-established (mel curve, sone curve, etc.), others are still waiting for discovery. It is one frontier of media understanding to implement more psychophysical results in retrieval and browsing systems.

23.4 Psychoacoustics and Psychophysics of Vision

Since the aural and the visual sense are of particular importance for media understanding, we dedicate the last section to the accumulation of psychophysical information about these senses. First, we deal with the auditory sense, then with the visual sense.

Psychoacoustics is of highest relevance for audio compression, but also for audio understanding. The fundamental *psychoacoustic model* includes the curve of absolute hearing thresholds by frequencies that has already been introduced in the first part. Other components are the curves for loudness perception (sone), pitch perception (mel) and critical bands (bark). The latter curve sometimes also serves as a – heavily quantized – measure for pitch/tonality.

According to [95], the *sensory pleasantness* of a sound is determined by *loudness, tonality, roughness and sharpness*. For the expression of loudness and tonality the sone and mel curves can be used. For roughness (in *asper*), the following model is suggested.

$$R = 0.3 f_m \sum_{i=0}^{24} \Delta L \tag{23.5}$$

Here, $\Delta L = 20 \log(\frac{f_m}{2})$ and i iterates over the critical bands. The modulation frequency f_m is the frequency of the hull curve of the audio signal. That is, roughness R is increased by complex sounds. The structure of this formula is similar to the one of information entropy. Signals with average frequency have maximal roughness while extremes (low modulation frequency, exceptionally high modulation frequency) will have low roughness, in the latter case because the individual sound components cannot be distinguished anymore.

Eventually, sharpness (in *acum*) is defined as follows.

$$S = 0.11 \frac{\sum_{i=0}^{24} i.L.g_i}{\sum_{i=0}^{24} L} \tag{23.6}$$

As before, i iterates over the critical bands. L is the loudness per band and g_i is a function that remains at unit size up to the 16th critical band and increases over-linearly afterwards. That is, the sharpness of a sound depends mostly on the relative loudness of the high-frequency bands. The higher the band, the more important. In summary, the scales of sensory pleasantness provide a sound foundation for audio feature transformation. Tonality and loudness are heavily used in audio understanding today. Roughness and sharpness should also not be neglected.

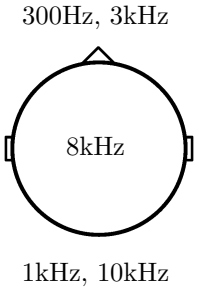


Figure 23.12: Frequency-Dependent Spatial Sound Perception.

Two further aspects of psychoacoustics require brief discussion here: *sound localization by binaural hearing* and *masking*. The influence of binaural hearing on sound localization is illustrated in Figure 23.12. There is a general tendency

to locate sounds at specific places depending on their frequency. For example, 300Hz sounds are generally perceived as rather coming from the front while 1kHz sounds will be perceived as having the source in the back of the subject. It may be worth considering this issue in feature transformation, for example, in environmental sound understanding applications (e.g. localization of car sounds).

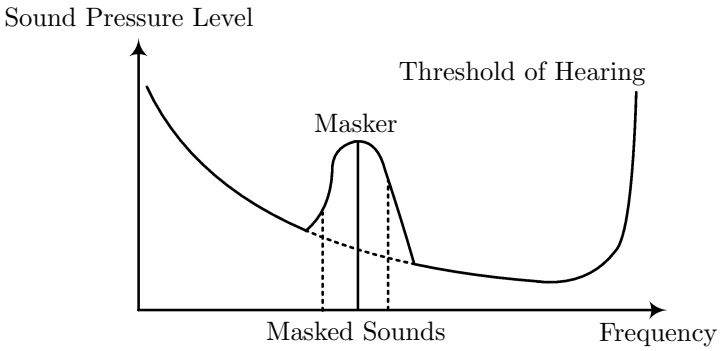


Figure 23.13: Audio Masking Example.

Eventually, masking of sounds is a problem of fundamental importance in psychoacoustics, in particular, in compression applications. We mention it only briefly, because it has – so far – only been of minor importance in media understanding. The principle is shown in Figure 23.13. A masking sound (masker) with high sound pressure level (SPL) changes (increases) the hearing threshold locally. Sounds with neighboring frequencies that fall under the new curve are masked, i.e. not perceived by the listener. The amount of increase of the threshold depends on the frequency and the SPL of the masker. Masking has been investigated in detail in psychophysics. The results are used in audio compression to neglect particular segments of the source signal. It appears reasonable to imitate human masking behavior in media understanding. For example, the music genre classification problem could probably be simplified by the exploitation of sound masking.

In the field of the psychology of vision we would like to limit ourselves to introducing some fundamental terms and pointing out their potentials for application in media understanding.

- *Colorimetry* investigates the relationship between physical light properties and human color perception. Aspects include the three-stimuli-theory, color models and the handling of color temperature. In particular the last aspect has potential for further exploitation in media understanding.

- *Visual acuity* measures the capacity of the human visual system for spatial

resolution, i.e. the clearness of seeing. Visual acuity is a property of the individuum. It is thinkable to implement this aspect by calibration in visual media understanding systems in order to provide a search tool that truly represents individual human perception.

- *Stereopsis* investigates human perception of depth by stereo vision and models it by depth maps. The entire research discipline is based on the visual differences (disparities) caused by the different locations of our eyes. The brain manages to create a spatial field of vision already in the early stages of perception. The imitation of this ability in media understanding would be desirable for all application domains that provide stereo data. For example, stereopsis and stereo cameras could improve the automatic detection of events in video surveillance (e.g. in patient monitoring systems) significantly.

- Eventually, the human visual system has a remarkable ability to analyze object motion. Hence, it appears worth trying to combine object similarity and movement similarity in visual media analysis. Such an approach would use coarse representation for the object shape (e.g. shape moments) but detailed methods (motion trajectories) for motion description. Application scenarios include sports video analysis and video surveillance.

We conclude that the psychology of perception is of highest significance for media understanding. Man is our measure. A media understanding system that ignores that, will fail. With respect to the fundamental issues of media understanding, psychophysics has a generally positive influence on the semantic gap problem, because it improves the context of the input data. Furthermore, there is a tendency that noise will be eliminated by psychophysical transforms.

This chapter and the last were concerned with human-centered problems. We investigated the influence of media on perception, cognition and high-level thinking. The results will – hopefully – help us to implement more human-like media understanding systems. The next two chapters are dedicated to this practical problem. We gather and investigate methods for the practical implementation of semantic template matching.

Chapter 24

Description by Templates

Revisits the fundamental convolution problem, links it to human similarity measurement, lists templates for audio, biosignals and stock data, and introduces static and dynamic models for visual media representation.

24.1 Convolution Everywhere

The *template* is the central carrier of semantic information in media understanding. Comparing the results of feature extraction for some media object to a given template results in a *belief score* for their similarity. *Identification* is a high belief score. Doubt is a low one. This chapter is dedicated to the template matching problem. Template matching is of paramount importance in feature extraction, hence, we already discussed various aspects of the problem and introduced several approaches in previous chapters. Here, we endeavor to give a systematic overview over the methods applied for template matching – independent of the underlying type of media. Template matching falls into two subproblems.

- *Representation* of the template
- *Similarity measurement* between template and stimulus

The first problem is a feature transformation problem. Templates are usually given as real-world objects, i.e. media objects of the same morphology as the stimuli (examples). Their semantic content has to be encoded in a description like for all other input media. The similarity measurement problem is typically solved by *convolution*. In the first two parts of this book, we have already

encountered *positive convolution* (similarity measurement, for example, by the dot product) and *negative convolution* (distance measurement, for example, by Minkowski distances). In Chapter 28, we will see that these forms of convolution are the end points of a continuum on which we can measure similarity – directly or as some form of generalized distance.

In the four sections of this chapter, we introduce a number of relevant methods for *template representation* and *template matching*. As it is the goal of this book, we make every effort to identify communalities and differences between the methods. Such communalities do exist, in particular, between media types with similar dimensional structure. Therefore, the chapter is organized by media types. After this introduction, the second section discusses template representations for audio, biosignals and stock data as well as for symbolic data (text, bioinformation). The last two sections focus on visual media: Section 24.3 on methods for representation based on statistics, Section 24.4 describes template building methods that use deformable objects.

Why are template representation and template matching frontiers of media understanding? Because of their outstanding importance. In fact, there are only two options to introduce context (semantic information) in the media understanding process: by ground truth and by templates. The first method has its limitations – as we discussed in several places. It is a very difficult, not to say almost impossible, undertaking to define a well-balanced, representative ground truth for most media domains.[1] The introduction of templates is, in comparison, straightforward. Hence, a unified theory for template representation and matching would be desirable. Unfortunately, this theory has not been defined yet. This chapter should help to push the semantic frontier of media understanding in this direction.

The remainder of this first section deals with the template matching problem. First, we state the general convolution model and add some psychological aspects of human similarity perception that will be discussed in detail in Chapter 28. Then, we introduce and cluster several techniques that were proposed for template matching. Some of these methods were already used in earlier chapters, others have to be employed together with specific template representations.

Interpretation is the central building block of most feature transformations. In the second part, we distinguished two types of interpretation operations: *autocorrelation* and *crosscorrelation*. The first compares one part of a media object to another. The second method compares the media object under investigation to some given template. Among the templates we discussed for feature transformation were moments (e.g. Zernike moments, Hahn moments, angular radial transform, etc.), base functions (angular functions, wavelet mother functions, contourlets, etc.), local media properties (visual keywords, gradient histograms,

[1] This fact may be seen as one form of Plato's problem (in concept theory).

etc.) and quantized references (e.g. bags of features). This non-exhaustive list makes clear how important crosscorrelation by templates is for the semantic interpretation of media content.

Crosscorrelation is implemented in media understanding as some form of convolution. For example, discrete transforms are usually based on convolution by the dot product. This form of crosscorrelation is a similarity measure. The result is maximal where the input media data match the template data. Other methods, e.g. histogram comparison, are typically based on convolution by distances functions. There, the result is minimal where input data and template data match. We named these two forms of crosscorrelation *positive convolution* and *negative convolution*. We can say that crosscorrelation is generally operationalized by convolution. Since crosscorrelation is one form of similarity measurement, convolution operators are similarity measures.

As the author could show in earlier work, positive convolution and negative convolution define a scale of similarity measurement on which all similarity measurement methods listed in the Appendices can be positioned. The two extremes of this scale are the dot product and the L_1 metric. All other measures can be expressed as linear combinations of these. For example, the statistical correlation coefficient and Tversky's feature contrast model lie both in the center of the scale. In consequence, any form of template matching is also a combination of the two extremes. See Chapter 28 for details.

Aspect	*Positive Convolution*	*Negative Convolution*
Notation	$A \otimes B$	$A \bar{\otimes} B$
Representative	Dot Product	L_1 Norm
Measure Type	Similarity	Distance
Stimuli	Separable	Integral
Thinking	Taxonomic	Thematic
Availability	Surface	Deep
Complexity	Low	High
Concept Theory	Classical	Prototype

Table 24.1: Aspects of the Two Fundamental Convolution Operations.

Hence, before we continue with concrete examples for template matching methods we consider it beneficial to investigate the differences between positive and negative convolution. Table 24.1 lists the most relevant aspects. The first three aspects should be clear by now. Concerning the fourth, psychologists have found out that positive convolution works best for so-called *separable stimuli*, i.e. countable properties and on/off-values. Distance measures perform superior for *integral stimuli*, e.g. the lengths of figures. Similarity judgment that is based

on separable stimuli is called *taxonomic* (e.g. comparing species in a biological taxonomy) while the holistic evaluation of the similarity of stimuli is called a *thematic judgment*. Similarly to the type of stimuli, descriptions suitable for positive convolution are sometimes referred to as *surface features* while integral stimuli are called *deep features*. The latter should be harder to comprehend, i.e. their complexity is higher than the complexity of separable stimuli. It is important to note that complexity is judged from the semantic point of view of human cognition here. That is, surface features are easy to comprehend for humans – but hard to extract for machines and vice versa. Eventually, there is an analogy between similarity judgment and concept theories. Separable stimuli can be seen as representing the conditions of the classical theory which could be operationalized by positive convolution. In a similar fashion, prototypes can be seen as integral descriptions. We conclude that positive convolution and negative convolution are the two fundamental forms of template matching and that templates are the containers for context in media understanding.

In the remainder of this section we deal with a number of matching techniques that were proposed for templates. Naturally, all of them are located on the level of the categorization micro process. In the first two parts of the book we already encountered a number of techniques for similarity measurement. We had the earth mover's distance for histogram comparison, the Hausdorff distance and the bottleneck distance for visual template matching. The Fréchet distance M8 in Appendix B.3 is similar to the latter two. It measures the largest distance between template and stimulus.

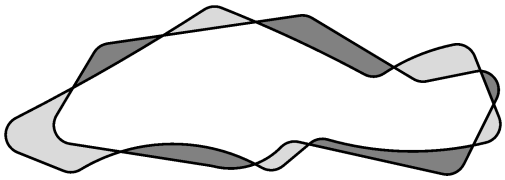


Figure 24.1: Template Metric Example.

Further micro-level measures discussed here are the template metric, the area of overlap and the elastic matching distance. The principle of the *template metric* is illustrated in Figure 24.1. The algorithm searches for the overlap between two objects – one being the template – that minimizes the gray areas. The resulting *area of overlap* is a measure for the similarity of two shapes, the sum of the gray areas is a measure for their distance. Please note that the template metric can be seen as the integral over the Hausdorff distance. Due to the aggregation, it is less prone to noise – paid by higher computational effort. The template metric fits naturally to visual template matching. However, it can also be employed for any other form of template matching. The principle is: vary the overlay until

the are of overlap is maximal.

The *elastic matching distance* is based on the idea of similarity meta models and of dual process models (see Chapter 28). For two descriptions f_x, f_y with properties x_i, y_j it searches for the minimal assignment with respect to stretching function s and distance m^{-1}.

$$m_{elmd}(x, y) = \min_{i,j} \sum m^{-1}(x_i, y_i) + s(x_i, y_i) \tag{24.1}$$

For visual objects (typically, curves), function s can be defined as the distance in length and function m can be defined by the inverse cosine measure. Then, the elastic matching distance sums up over a similarity measure (cosine) and a distance measure (e.g. city block metric), which is the requirement for a dual process model. Furthermore, the elastic matching distance has a signature very similar to the Mallow's distance. The major difference is the usage of summation instead of multiplication. Hence, the elastic matching distance should be more forgiving for suboptimal data.

The common problem of the presented similarity meta models for template matching is their computational complexity. Most methods require the identification of properties with minimal/maximal distance (e.g. neighboring edges), some even twice. Furthermore, template matching is cursed with a number of degrees of freedom: position of the template, scale, and some semantic aspects (e.g. those discussed in Chapter 12 for semantically meaningful wavelet mother functions). The dimensionality problem is a typical side effect of hot media, such as semantic templates. General simplification approaches that were suggested are *subdivision* of the search space and *parameter interpolation*. Search space simplification can be reached by presenting only a limited number of options per dimension. Parameter interpolation aims at the elimination of entire dimensions by predictive coding through other dimensions. Factor analysis methods can be used to reach this goal.

In conclusion, the usage of context in the form of templates requires solutions for their representation and for matching. Matching means convolution, mostly by similarity meta models. Strategies for efficient representation are discussed in the remaining sections of this chapter.

24.2 Templates for One-Dimensional Media

The one-dimensional data types discussed here are audio, biosignals and stock data. Bioinformation and text – the symbolic data types – are only briefly considered. All of the quantitative data types have the same basic model of template matching in common. It consists of the following three steps.

1. Signal smoothing

2. Template preparation

3. Template matching

The third point has already been discussed in the last section. The matching methods applied on one-dimensional data are not different from those applied on visual data. *Signal smoothing* is the necessary prerequisite of template matching for one-dimensional data. It would hardly make sense to apply a template directly on an audio stream or a biosignal: In audio, the sample rate is too high and – as we could show in the second part – the sound information is distributed over all samples. Biosignals, on the other hand, have a too bad signal-noise-ratio, i.e. template matching directly on the signal would be biased significantly by the noise component. Even stock data (few samples, no noise) benefit from smoothing as the popularity of sliding averages in technical analysis indicates.

Signal smoothing is typically achieved by averaging. Averaging can be performed for static or sliding windows by any statistical moment of first order. Frequently, the mean is used since – despite all noise – the media types under consideration usually do not contain strong outliers. In the audio domain, static windows are employed more often than in the biosignal and stock domains where sliding average methods are prevalent. The result of signal smoothing is a *hull curve* that describes the shape of the signal over large spans of time.

The remainder of this section is dedicated to the second step of the basic model: template preparation. We discuss the fundamental types of templates that have been employed successfully on media data. First, we focus on the audio domain, then biosignals and, eventually, stock data.

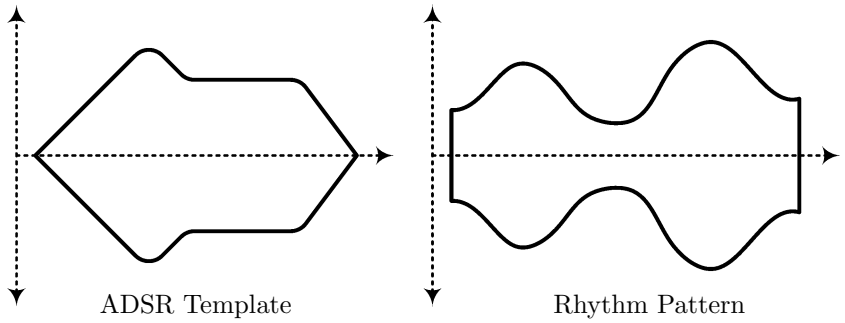


Figure 24.2: Audio Template Examples.

In the audio domain, we have already encountered a number of templates based on the hull curve. The left part of Figure 24.2 illustrates the *attack, decay, sustain, release* model (ADSR) of sounds that is, for example, implemented in the MPEG-7 log attack time descriptor. According to this model, every sound

that is part of an audible sensation consists of a strong fade-in component, a short decay, longer sustain and, eventually, a fade-out. The hull curve of an isolated sound can be described well by this model. All that is required is adaptation to the template which is usually implemented by some similarity meta model and an acceleration technique such as parameter interpolation.

In the last chapter we defined the roughness of a sound by the modulation frequency, which is the frequency of the hull curve. This high-level description of an audio sensation may also be seen as an application of template matching. Generally, we can use the hull curve to measure the fundamental frequency of a sound. Coarse estimation of the fundamental frequency can be reached by the sequential convolution of the hull curve over frequency templates. The one with the extremal matching score receives the highest belief.

In a similar fashion, template matching can be used to detect arbitrary rhythm patterns. For example, the right part of Figure 24.2 describes a sinusoid pattern. In the same way, pulse and beat patterns can be defined and measured by template matching.

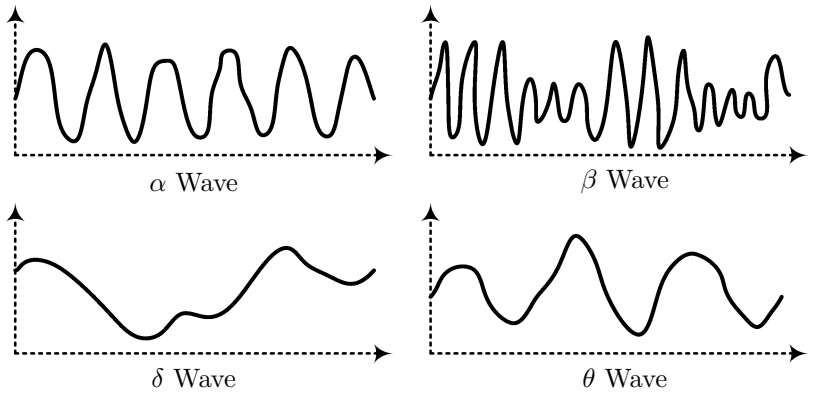


Figure 24.3: Waveforms of Biosignals.

In the biosignal domain, it makes sense to derive the templates for hull curves from the fundamental wave types. Four wave types are illustrated in Figure 24.3. The α wave with a typical frequency of 10Hz is a result of relaxation and closed eyes (not sleep). Children show α frequencies below 8Hz. The β wave has a frequency around 25Hz. In contrast to the α wave which can most easily be captured from the posterior regions of the head, β waves can be captured in the front part. Such a wave indicates (anxious) thinking or movement of the subject, rhythmic β waves indicate drug effects. The δ wave has a frequency around 3Hz and the highest amplitudes of all wave types. This wave indicates a sleeping subject (adult or baby). It can be measured at the front and the back

of the head. Eventually, θ waves have a frequency around 6Hz and are typical for children. Meditating adults also show θ waves, where it may indicate arousal as well. Furthermore, θ waves indicate several mental disorders.

For biosignals with higher frequencies (e.g. γ waves of 100Hz) it is recommendable to generate a hull curve for matching by smoothing over ten samples or more. For the detection of low frequency waves it may even make sense to perform the convolution with a template that represents an idealized wave (without smoothing). This approach makes sense for α waves and θ waves. The irregularity of δ waves, however, makes this signal type a hardly suitable ground for template matching.

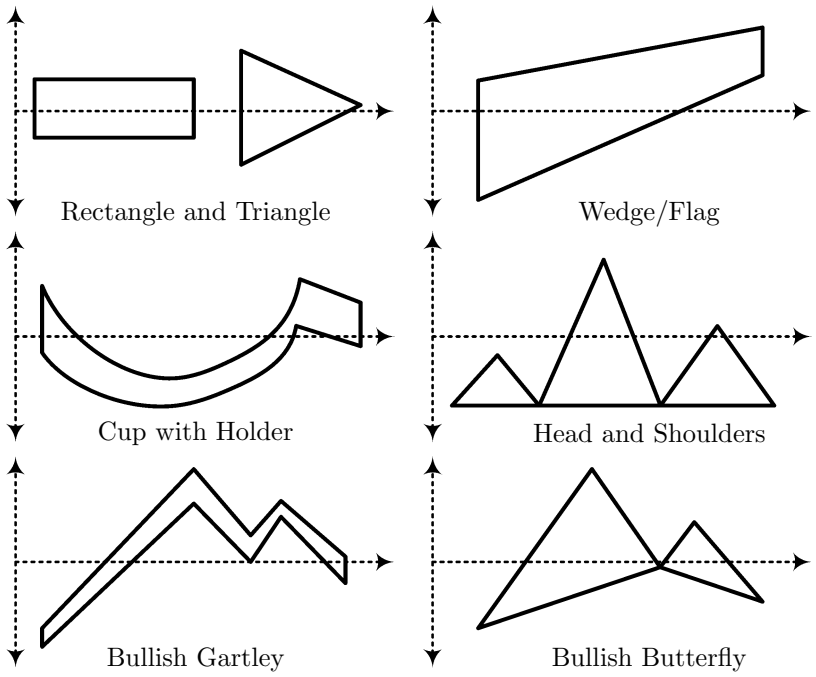


Figure 24.4: Chart Template Examples.

The stock domain is very interesting for template matching. Such methods have been used in technical chart analysis for decades. Unfortunately, this has been done with inferior strictness – 'observing' a particular pattern wherever desired. The methodology of media understanding, in particular, the more rigid categorization techniques could turn stock data template matching into a less arbitrary, more successful prediction method.

Figure 24.4 illustrates the hull curves of seven important stock data patterns. Like for the biosignals above, we explain their meaning in the subsequent paragraphs. Their application is straightforward. Since stock data has an exceptionally low bandwidth (often, one sample per day), we can even skip the signal smoothing step (or apply a sliding average). The rest is template matching by negative convolution (frequently used method) or positive convolution (also interesting, if notations are interpreted as separable stimuli).

The upper left diagram of Figure 24.4 shows a *rectangle* and a triangle. The first pattern stands for a share that moves within a support and a resistance line. The smaller the vertical component of the rectangle, the more constant (predictable) the share. The triangle indicates a market of increasing insecurity. Trading volumes are decreased until the market escapes the triangle. The *flag* in the first row of the figure indicates an increasing market from which some of the bears (pessimists) extract their contribution while more bulls (optimists) invest. The flag is one of the most common templates in chart analysis. Theory says that at the end of the flag the course of the share would escape into the direction of the flag (here: up).

The second row of the figure shows a *cup with holer* and the *head and shoulders* pattern. A cup with holder is typically located after a longer period with varying exchange volume and before a boom/depression. The left part of the cup stands for bearish consolidation and the right for re-investment. The holder covers the time span in which some bears extract their profit. Head and shoulders is a typical reversal patterns. The example shows the end of a positive trend and its reversion into a negative one. The two shoulders are produced by bearish/bullish behavior on small scale (left: indicating, right: reacting) while the head represents the actual turn of the market.

The two templates in the bottom row of Figure 24.4 stand for reversal patterns as well. At the end of the *Gartley pattern* the trend should move upwards. The first peak serves as an indicator for the later boom. The *butterfly pattern* is similar to the Gartley, also indicating a later boom. The major difference is here the grater variance in the pattern which reduces its belief score.

We would like to emphasize the two major shortcomings of these patterns in technical chart analysis again. Firstly, they are all based on partially dubious experience. Most of these patterns were suggested by individuals based on their experience and not as the result of quantitative analysis. It would be desirable to investigate the patterns of stock data quantitatively and to develop indicator patterns that are not influenced by theories about bullish/bearish behavior. Secondly, these patterns are used very generously in technical chart analysis. Cups with holder, for example, can be interpreted into almost any curve. If these indicators should be meaningful, they would need to be applied in stricter form – of course with the disadvantage of fewer hits.

There are hardly any templates for the identification of symbolic media data. Exceptions are certain gene patterns that can be recognized by structural alignment – which may be seen as a form of a similarity meta model. In the text domain, grammar models are one exception. However, these templates are located on a meta level and serve rather as models for patterns.

In conclusion, template-based understanding of one-dimensional media objects is a sequence of smoothing, representation and matching. For audio, biosignals and stock data many useful templates do exist that can be employed for the extraction of semantically meaningful descriptions.

24.3 Static Visual Templates

This section and the next introduce template methods for visual content. Due to the visual connotation of the word *template* it is not surprising that a great number of visual methods has been proposed. Some of these methods are so specific that template representation and template matching cannot be separated, or, that the methods make only sense for the type of representation. Hence, we explain the matching methods – where needed – in place. Furthermore, there is often no smoothing in visual template matching. Of the two sections, the first focusses on static template matching, i.e. the adjustment of the data to the template. The next section focusses on dynamic template matching, that is the adaptation of the template to the data.

In this section, we order the methods by *decreasing model complexity*, which is a generally desired goal in template matching. Where possible, we compare and group methods, even though this can – due to the heterogeneous nature of the templates – seldom be performed. Visual templates can appear in various forms. In the first two parts of this book we already encountered visual keywords (simple approach, complex template), discrete transforms based on angular functions (complex approach, simple template), contour/edge-based methods, trajectories, etc. In these sections, we add complex methods such as angular signatures, scale spaces based on curvature, active contours, etc. In particular, below we discuss three types of visual templates.

- View-based templates
- Point-based templates
- Curvature-based templates

View-based representations are straightforward. Figure 24.5 shows an example. The object of interest is represented by expressive views. It goes without saying that view-based methods can easily be applied on 3D model information

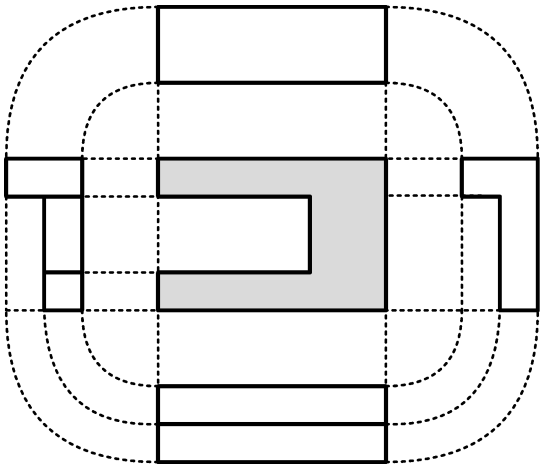


Figure 24.5: View-Based Object Representation.

but only with great difficulties on pixel-based image and video data. There, the view can be a set of visual keywords or a bags of features description. MPEG-7 provides a descriptor for view-based templates. The *2D/3D Shape Descriptor* is able to capture a 3D model together with its views. Since the views will usually be redundant the selection of views can be considered a minor problem: every view that helps the media understanding process should be included.

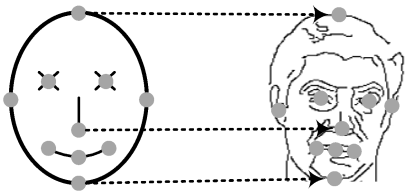


Figure 24.6: Point-Based Object Representation (© *CNBC*).

Figure 24.6 shows an example for a point-based face template (left) and the matching with a media object. In the example, we use manually defined feature points and compare them to mechanically detected keypoints based on edge curvature. Generally, point-based templates can be described as *super-local descriptions*. Local features are aggregated and clustered. In the second part, we already stressed the importance of object boundaries for the expressive description by local features. Here, we use the opposite argumentation for reaching the same goal. Interest points can very well be used to formulate templates of

objects. The suggested approach of comparison of template point sets to extracted keypoints depends in its success only on the quality of the interest point detection method. Every method that is not exclusively based on curvature but as well on repetitive patterns (e.g. the Gestalt laws suggested in Chapter 14) can be employed for this purpose. The general process will always consist of the following steps.

1. Preparation of a point-based object representation
2. Application of a liberal interest point detector
3. Clustering of neighboring points
4. Template matching

The clustering step serves as smoothing and quantization of the point space. For the matching step, for example, the Mallows distance can be used – but as well one of the other similarity meta models.

The group of curvature-based template representation methods is related to the point-based ones. After all, point features are usually extracted based on high curvature. The major difference lies in the *direct* application of curvature information here. For illustration, we introduce three methods: two based on angular information and one scale space.

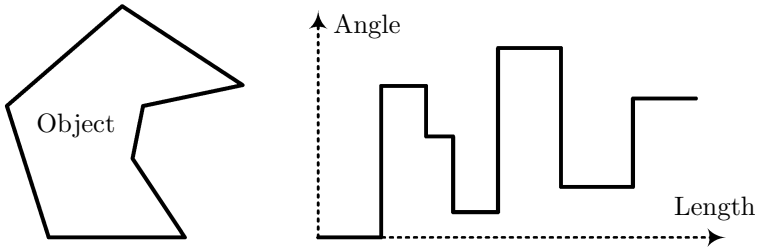


Figure 24.7: Turning Function Example.

The principle of the *angular turning function* is laid down in Figure 24.7. The object/template contour is transformed into a signature of edge lengths and the angles between pairs of edges. Matching can, for example, be performed by dynamic warping. Since dynamic warping is a rather complex and resource consuming procedure the signature can also be simplified by neglecting the length of edges and comparing the angular signatures by a distance measure. In contrast to the full signature, however, the angular signature does not carry the entire object information anymore. For further simplification, the angular signature can be reordered or quantized into a histogram.

We see that the template representations become simpler from method to method. Views are full visual objects, point sets approximate the visual information, turning functions represent the object outline by a signature. The *shape context* method reduces the model even further and builds a histogram. The algorithm takes the following steps.

1. Detect the edges of the visual template/object.
2. Compute a *shape context histogram* by the following steps.
   (a) Select n points from the edge representation.
   (b) Compute length and magnitude of the local gradient from each selected point to each other selected point.
   (c) Build a histogram of the logarithmic polar coordinates of the gradients. The result is a matrix of angles and distances.
3. Transform the histogram to an array g by a zigzag scan and normalize g by the mean.

Principally, any similarity or distance measure can be applied to compare two arrays. The authors of the shape context model suggest a complex distance function, in which the χ^2 characteristic of the two arrays is the central element. For two arrays x, y and a free parameter a it takes the following form.

$$d(x,y) = a\frac{1}{2}\sum \frac{(x_i - y_i)^2}{x_i + y_i} \tag{24.2}$$

This term measures the quality of the match on the level of the points. Other terms of the distance function measure differences in brightness, costs of transformation, etc.

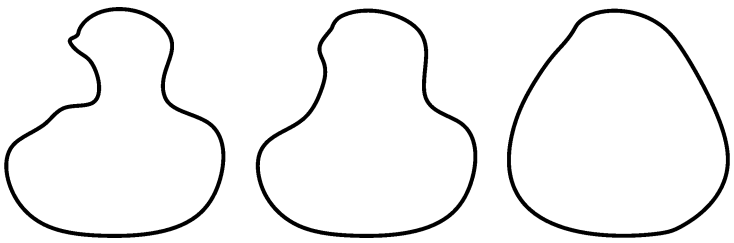


Figure 24.8: Curvature Scale Space Example.

The last method to be discussed in this section is the *curvature scale space* as it is, for example, used in the MPEG-7 *Contour-Based Shape Descriptor*. The

idea – implementation of the scale space principle for the curvature of object outlines – is illustrated in an example in Figure 24.8. The scale parameter t controls a Gaussian smoothing process that is applied on the angles between edges and the flexion of edges. Over t, complex shapes are reduced to geometric primitives that can easily be described by shape moments, the simplest form of template representation. For the matching, an iterative procedure is suggested: from the coarser levels of representation towards the more detailed ones. In summary, this method is very similar to the application of scale spaces for the representation of point features.

All of the presented methods are practically useable. They have in common that a static representation is computed from the template content. The scale of methods allows for selecting the suitable technique for a particular media understanding problem with respect to model complexity and redundancy. Generally, a higher degree of model complexity and informativeness will be paid with higher redundancy and higher resource consumption. This decision cannot always be left to the system designer, in some cases it must be taken ad hoc. Then, the dynamic template models discussed in the final section of this chapter are of interest.

24.4 Dynamic Template Adaptation Models

This section extends the brief introduction of *energy-based contour models* in the first part of the book. There, we gave an example of an *active contour* for the description of an object outline. In this section, we describe the underlying model in greater detail, explain the process of model adaptation and, eventually, the matching procedure between stimulus and template. As already mentioned, the active contour approach takes the direction opposite to statistical template representation. Here, we adapt a model to given data, there the data to a given template. Hence, active contours are dynamic models with all advantages and disadvantages, in particular, higher flexibility and the risk of overfitting and suboptimal solutions.

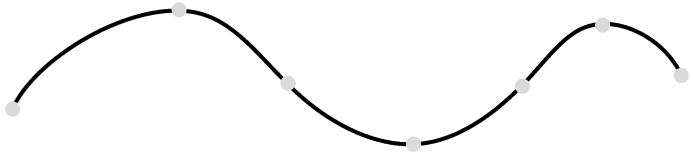


Figure 24.9: Spline Example.

The essential building block of an active contour (or, *snake*) is the *spline*. Figure 24.9 shows an example. Mathematically, a spline is a polynomial function

that is defined by control points. The curve is smooth for the first few derivations. For object representation, we have as a necessary requirement that connected spline segments must have smooth transitions. The spline for given control points can under certain conditions be found analytically or – as often – by a heuristic procedure, e.g. expectation maximization.

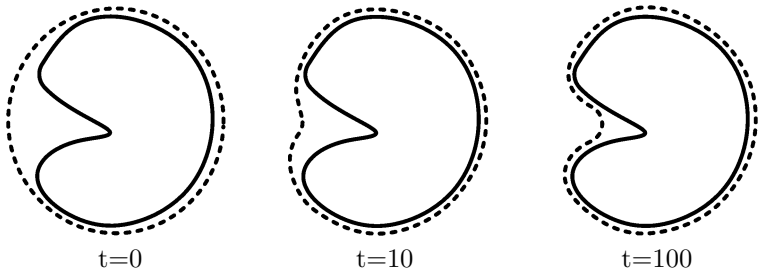


Figure 24.10: Adaptation of an Active Contour over Time.

One or a group of splines is the basis of every snake. The snake is used to describe the contour of an object – in our case a template (e.g. given as an edge map). Essentially, the snake is the spline model that fits best to the given object outline. What *best* means here is discussed in the next paragraph. Figure 24.10 shows an example of a snake. The time parameter t indicates that finding the snake for the given splines is an iterative optimization process. With increasing t the spline template fits better and better – however, not beyond a certain degree – to the object contour.

The evolution of this process is steered by the template matching function, which is typically defined as follows.

$$e = \sum e_{int}(f_i) - e_{obj}(f_i) \rightarrow \min \tag{24.3}$$

That is, the total energy e for a given spline $f_i = (x_i, y_i)$ is the sum of two (in some models, even more) components. The first stands for the *internal energy* (i.e. resistance of the model against deformation) and the second for the *object energy* (i.e. the degree of adaptation to the input data). Internal energy should be minimal: the model should be deformed as little as possible. Object energy, on the other hand, should be maximal in order to capture the outline of the object well. The tension caused by the opposite directions of these two energy components creates the optimization problem of the active contour. The internal energy can be defined in the following way.

$$e_{int} = w_e \left| \frac{\partial f}{\partial i} \right|^2 + w_s \left| \frac{\partial^2 f}{\partial i^2} \right|^2 \tag{24.4}$$

Here, w_e is a weight for elasticity, w_s is a weight for stiffness. The internal energy is increased over-linearly by a bent spline (first term) and by variable curvature (second term). The configuration of the two weights determines which aspect receives the higher penalty. In the simplest form, object energy can be defined as follows.

$$e_{obj} = w_{line} \sum_i o(l_i) \tag{24.5}$$

Here, $o(l_i)$ is the gray level of the input object at location l_i, i iterates over all media samples that are covered by the snake, and w_{line} is a weight the defines the attractive side of a contrast. Depending on the brightness scale, a positive weight will stand for attraction to dark pixels and vice versa. Practically, the line term in object energy is supplemented by components for edges (e.g. modeled by gradients) and connection points. However, the basic idea remains the same: *minimal deformation at maximal quality of fit.*

Active contours perform some sort of negative convolution (quality of fit) while taking certain noise components into account (e.g. stiffness term). It is one nice aspect of the model that depending on the needs, optimization criteria can be added or removed. The optimization process is usually implemented by some dynamic programming algorithm. Any global optimization procedure – including those discussed in the second part – is applicable.

In conclusion of this chapter, template matching is a crucial building block of feature extraction for media understanding. Templates contribute to a higher semantic level of the media understanding application and less polysemy – paid with higher dimensionality of the description problem and worse computational performance. Its importance and the diversity of the applied methods makes template matching a frontier of media understanding. We believe that future efforts will see a unification of template matching methods based on the three-step procedure of representation, smoothing and mapping, where the latter means one or another form of convolution. We are positive that what is actually required is not more research on representation methods but a unified theory of similarity/correlation/convolution. Chapter 28 contributes to this end. Before, however, the next chapter builds semantic applications on the introduced template matching methods and other feature transformations.

Chapter 25

Semantic Descriptions and Applications

Introduces the semantic scale, describes the usage of low-level descriptions for semantic enhancement and semantic applications in the audio and the visual domain.

25.1 The Semantic Scale

Media understanding applications need to be *semantic*. What does that mean? It means that the application has to judge the content of media objects *in the same way* as humans do. Neither should the application fall behind human reasoning, nor should it be ahead of our perception apparatus. Judging the content means recognizing objects, patterns, templates, spatiotemporal relationships, etc. – the entire domain of perception and cognition.

This chapter is dedicated to the discussion and reflection of the ambitious goal to find the right level of semantics. The central element is presented in the first section: the semantic scale. It should sensitize the reader with respect to the fact that the ideal semantic level is a fragile thing. What the machine should imitate and to which extent depends on the position of the user, her views about the media domain, etc. Media-theoretic aspects are involved as well as the level of expertise in the usage of media understanding systems and many other factors. It is obvious that for being hidden and not retrievable the large majority of these properties cannot be considered in media understanding applications. Some, however, can. These are discussed in the subsequent sections

of the chapter. The second section deals with low-level feature transformations that can be contextualized to some level of semantic understanding. On top of these and earlier introduced feature transformations, the last two sections review some state-of-the-art solutions for semantic media understanding domains of outstanding practical importance, namely face recognition, speech recognition and emotion recognition.

This chapter builds on earlier introduced thoughts. In particular, in Chapter 11 we discussed the term *context* as the operationalization of semantics in media understanding. The context of an application determines how media content has to be interpreted. We said that time and location play an important role. So will the genre for videos. It is a difference, if the anchorman of a newscast or a comedian makes a joke.[1] The context provides the direction for the information filtering process implemented by feature transformations and classifiers. The transformation of media samples into semantically loaded class labels is influenced by a multitude of factors. For example, we introduced the application of context in categorization. There, we have the *ground truth* as the fundamental form of context. Ground truth is world information that defines the meaning of media patterns. For feature extraction, we discussed the influence of Gestalt laws on the perception of local features. This thought is prolonged in the present chapter. In the second section we introduce feature transformations for capturing symmetries and self-similarities in visual media objects. Among other purposes, these descriptions provide the ground for the application of Gestalt laws.

This section is dedicated to two theoretical issues of semantic media descriptions and applications. First, we define the semantic scale for the localization of media understanding efforts. Then, we discuss the chances and risks of defining tailor-made application for – sometimes, narrow – media understanding application domains. Since such applications are the topic of the last two sections of this chapter, we consider it beneficial to clear the fundamental problems bound to this scheme beforehand.

Where do semantic descriptions and semantic applications stand with respect to concept theory? The question is relevant since, eventually, semantics and context stand for the desire to load signs with strong denotations. From the theoretical point of view, the question cannot be answered. The concept-theoretic approach depends on the semantic problem domain. In practice however, we can say that *theory theory* triumphs. Semantic descriptions are the result of iterative media understanding processes where everything is taken into account that helps the – often, slow – enrichment of descriptions and (proto-)predicates. We may state this as a general rule: Semantic enrichment will make use of whatever knowledge is available about the views of the user and of whatever methods are

[1] In the latter case, it has to be better.

available for the employment of this knowledge. Hence, semantic enrichment will take place in feature extraction, categorization and refinement. We discussed semantics in the earlier machine learning chapters – categorization methods cannot be explained otherwise. Hence, the focus of this chapter is on the feature transformations and on the applications composed of semantic transformations and semantic categorization. We are positive that the semantic application is the future of media understanding. From a media-theoretic point of view, it could be argued that *the media is the message* implies *putting the human in the loop* which can only be performed successfully in iterative applications, not in static transformations nor in one categorization cycle.

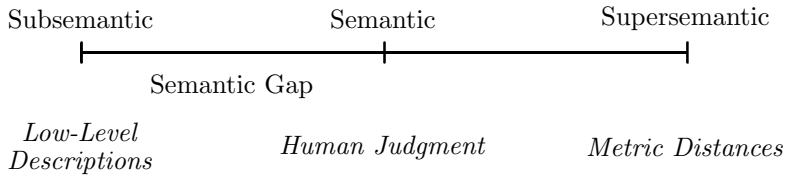


Figure 25.1: The Semantic Scale.

Figure 25.1 introduces the semantic scale of media understanding. In the center we have the desired – yet hard to define – level of semantics. The center point – also the center of gravity of media understanding research – is surrounded by two wings. The left wing covers what we call *subsemantic* methods, i.e. methods that do not come up to the human standard of perceptual cognition. Between subsemantic and semantic lies the *semantic gap*. In the literature, the semantic gap is often seen as a problem of feature transformations that are too simple for representing human *high-level* perception appropriately. And really, the best examples for subsemantic methods come from the feature transformation domain. The zero crossings rate, however simple and useful, is clearly a subsemantic representative of pitch/tonality. A color histogram is certainly an inferior color description compared to human visual memory. The head and shoulders pattern is only a simple operationalization of the complex interaction of bulls and bears on a market, and so on.

The right wing is the new idea in the semantic scale. *Supersemantic* methods are those that are clearly better than human perception – for example, audio feature transformations that do not suffer from the insufficiencies of human hearing. From a purely logical point of view it appears surprising why we should consider supersemantic methods inferior to semantic ones. Human-centered evaluation, however, makes quickly clear that the vast majority of users prefers semantic results because they can *comprehend* such results whereas supersemantic results may appear interesting sometimes but, often, are just misunderstood or not understood at all. After all, evaluation in media understanding is based on human

ground truth, not a machine-generated ideal ground truth. As already stated a couple of times in the third part of this book, *man is the measure* in media understanding. Human judgment is what we define as semantic on the semantic scale. *Supersemantic behavior is as undesired from a system as is subsemantic behavior.*

Today, typical examples for supersemantic methods hardly exist in the feature transformation domain, because psychophysical results have already found application there. Insufficiencies such as masking, spatial distortion, nonlinear color perception, cognitive and statistical illusions are imitated in description methods by heuristic scales and transformations. In the categorization domain, however, there is significantly less sensibility for this issue, which is surprising if we consider the declared goal of machine learning: imitation of human learning. For example, the metric distances frequently used in the vector space model and related classifiers (k-means, k-nearest neighbor, self-organizing map) have been proven inadequate representatives for human similarity judgment. Some aspects of similarity measurement for integral/quantitative descriptions are not covered by the metric distances and some aspects covered by the underlying metric axioms (for example, the triangle inequality) are not human-like. Metric distances are perfectly logical, but unlike human behavior. Hence, they can justly be called supersemantic.

We conclude that feature transformations tend to be subsemantic and categorization methods – in particular, the micro processes – tend to be supersemantic. Dealing with subsemantic feature transformations is the topic of the next section. Dealing with supersemantic categorization methods is the topic of Chapter 28. Assembling subsemantic descriptions and supersemantic classifiers to semantic applications is the topic of Sections 25.3 and 25.4.

However, before we continue with turning subsemantic feature transformations into semantic ones, a second issue requires discussion. Solving certain semantic media understanding problems by tailor-made applications is an important area of media understanding research. These applications push the practical frontier of media understanding. Typical semantic domains are face recognition, speech recognition, emotion recognition, violence detection, music genre classification, etc. The common advantage of tailor-made applications is their *practical relevance.* Face recognition *is* required for a large number of real-world applications. Speech recognition is ubiquitous. Violence detection for surveillance systems would be highly desirable. The common problems of tailor-made applications are that they require ground truth (hard to define, hard to maintain, legal issues, etc.) and the *danger of over-narrowing the domain.*

With over-narrowing of the domain we describe the following phenomenon of media understanding research. The experimenter defines a narrow context for his applications – often, by a small ground truth data set, sometimes even the other way around – develops tailor-made feature transformations with 'magic' quanti-

zation steps (e.g. heuristic weighting) and chooses a categorization method with strong potential for overfitting (e.g. a random forest). Evaluated by the narrow ground truth, the tailor-made application will 'solve' the semantic application problem. However, the solution will remain irrelevant and the frontier unaltered if the domain has been narrowed too far: beyond practical relevance. For example, a speech recognition system that captures only a few words is practically irrelevant. A music genre classifier that covers only the work of one artist will not be of much use, etc. Over-narrowing is the case where the practical relevance of the semantics are not given. It is indicated by small ground truth sets. Over-narrowing applications are of little use.

In conclusion, the semantic scale should sensibilize the reader for two gaps: the subsemantic one, mostly caused by insufficient feature transformations, and the supersemantic one, primarily caused by categorization methods. In the next section, we review methods that try to improve subsemantic feature transformations for the better imitation of human behavior.

25.2 Semantic Feature Transformations

The feature transformations that we presented in the first two parts of the book were hardly ever on a semantically high level. The mel frequency cepstral coefficients (MFCC), for example, that play such a prominent role in speech recognition and all other audio understanding applications, are on a fairly low level. The entire recipe consists of template matching with a sine function, psychoacoustic quantization and decorrelation by the cosine transform. None of these elements would have a semantic meaning on the level of human cognition. There is, though, semantics in the MFCC, not in the transformation, but in the usage. The extracted pitch information has a semantic meaning for the listener of a piece of music.

The methods introduced in this section follow the same idea. The features themselves are rather simple but meet a semantic category of human life. This way, we come in reach of the ambitious goal to represent semantic information by descriptions. Below, we discuss feature transformations capable of representing semantics in the area of audiovisual perception. Transforms for other media types are discussed together with their applications in the next section. In the visual domain, we focus on the intelligent representation of color information and the semantic interpretation of contrast as symmetries and self-similarities. In the audio domain, we focus on one – due to its nature – often neglected, yet important aspect of audible sensations: silence.

Furthermore, we consider one particular type of description in this section that represents the biggest semantic gap: *random numbers as arbitrary descriptions.* In earlier work, the author could show that such descriptions may be used

in a highly semantic way, if enriched by intelligent feature selection methods or the description optimization kernel functions discussed in Chapter 18.

The first feature transformation on our list is a *silence descriptor*. For example, silence can be extracted as short-time energy relative to a minimal loudness threshold ϵ_l. Whenever the average energy for a time frame falls below ϵ_l the segment is considered to be silence. Why is silence a semantically relevant description? Silence is in many respects typical for certain individuals and applications. For example, the mixture of speech and silence can very well be used for speaker identification. Furthermore, silence is an interesting indicator for the segmentation of environmental sounds. For these applications it is as the proverb says: important is not what is being said but what is not being said. Heinrich Böll transformed this idea into a short story: *Murke's Collected Silences* (available in [38]).

In the visual domain, a first context for the application of low-level feature transformations is the recognition of paintings. There, low-level yet semantic color descriptions are of highest usability. For example, the *Itten color wheel* can be used to describe image content with respect to *pure colors* – a similar idea to the representation of pure tones in the Fourier transform. The Itten color wheel distinguishes 18 colors. It starts with pure red, yellow and blue, derives green, orange and purple from the pure colors and uses these six basic colors to define further six gradations, for example, light orange between yellow and orange. The color circle follows the direction of the rainbow: yellow, orange, red, purple, blue and green. Interesting aspects of the color wheel – developed in the context of the Bauhaus – are that it neglects human visual perception of colors. Green is only represented by two colors, red by four colors. This color model gives an interesting semantic re-weighting of the importance of colors in modern societies (and their media).

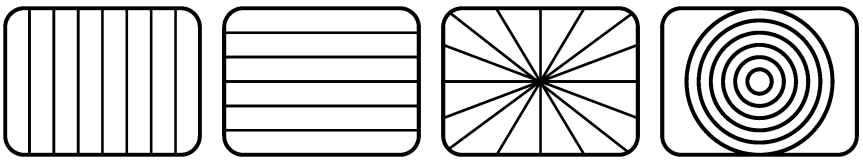


Figure 25.2: Kansei Color Composition Templates

Another interesting semantic set of low-level features are the *Kansei features*. Here, we focus on the color descriptions. The central idea is that Kansei color histograms should represent the feeling expressed by visual works. Therefore, they are aggregated over particular regions using a dominant color approach. Figure 25.2 shows the four basic composition templates. For each of the depicted regions, the dominant color is computed as the average along a perception-based

color model. Then, the color histogram is constructed from colors of the regions. The authors of the Kansei approach suggest specific similarity measures for their descriptions. However, we consider the template-based approach the major innovation in Kansei features. Like the MPEG-7 color structure descriptor, Kansei features are able to capture color semi-locally. We believe that the four presented templates should be supplemented by the typical patterns of application domains such as family photos, video shot types, etc. The combination of low-level color information and semantic templates has certainly potential.

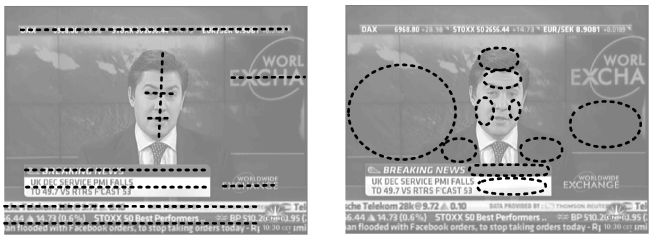


Figure 25.3: Symmetries (left) and Self-Similarities (right, © *CNBC*).

Symmetries and *self-similarities* are of highest significance for human visual perception. Unconsciously, we recognize several types of symmetries in objects and use this information to classify objects and to eliminate redundancy. The laws of Gestalt that were already mentioned several times are a practical consequence of this sense. Figure 25.3 shows a practical example. The left image shows some symmetry axes in the leading example. Symmetries can be detected in the face, clothing, background, text (e.g. verses), etc. The right side of the figure shows some self-similar areas. Some are trivially unicolor, while others show the same texture and shape. The visual example should make clear that these image properties with partially high-level semantics should be extractable by low-level feature transformations.

A straightforward approach for the detection of symmetries is the usage of edge information and of ridge detection. These methods should be able to extract symmetry axes along contrast lines, but not within objects. Within objects, we suggest the following technique. Arbitrary symmetry axes in visual objects can be extracted by usage of the *Lie group SO(2)*. For angle ϕ, the rotation group takes the following form.

$$SO(2) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \tag{25.1}$$

A simple symmetry feature transformation for given objects could be composed of the following steps.

1. Center the object at the point of gravity.
2. For all symmetry axes described by an angle ϕ of interest, perform the following operations.
   (a) Perform the rotation using SO(2).
   (b) Compute a score for every line of the object by taking the contrast in image intensity of the left and the right side of the symmetry axis.
   (c) Assume symmetry, if the sum of all line scores remains below a predefined threshold.

Alternatively, the Radon transform can be used to detect symmetry as invariance of the spectrum against 180 degree rotation. In fact, the suggested scheme is very similar to a Radon transform.

Self-similarity is operationalized by autocorrelation, i.e. template matching between the entire object and its parts. A self-similar feature transformation can be based on texture features or on the semantic aggregation of interest points. For example uniform distribution of interest points over a segmented area will indicate high self-similarity.

Another idea that appears promising is to base a self-similarity feature transformation on fractals, in particular, on *iterated function systems* (IFS) for the construction of self-similar patterns. The general approach is to find an IFS algorithm that is capable to describe the region supposed to be self-similar. In the past, such approaches have, for example, been used in image compression. We suggest the following algorithm for self-similarity detection in a visual object o.

1. Define the archetype of the self-similarity pattern $o_0 = T_0(o)$ by image size reduction T_0.
2. Apply the contracting image transform $o_i = T_i(o_{i-1})$
3. If the distance $d(o_i, o) \geq \epsilon$ return to Step 2

The idea behind the first step is that if an object is self-similar than a smaller version would be a fair representative for the whole. Therefore, we shrink the image in order to provide a starting point for the IFS. The contracting image transform has to satisfy the condition $|T_i(o_1) - T_i(o_2)| < a|o_1 - o_2|$ for arbitrary objects o_1, o_2 and a contraction parameter $a \in [0, 1]$. Typically, the contracting transform will be composed of the following steps.

1. Image size reduction of the input object
2. Duplication of the object

3. Rotation of one instance

4. Merging of the two instances

This recipe allows, for example, the construction of self-similar structures such as *Barnsley's fern*. In visual media understanding, the archetype plus all transforms provide a description of the self-similarity of object o. A self-similar object will neither have an IFS with very few transforms (indicates unicolor) nor with a very large number (indicates missing contraction, i.e. missing self-similarity). Hence, the number of transformation steps is already an indicator of the self-similarity of an object.

Eventually, the distance function d will usually be based on the attractor o.

$$d(x, y) = \inf(\alpha | x \subset \alpha y \wedge y \subset \alpha x) \tag{25.2}$$

Here, αx denotes object x with a border of width α. The distance of two objects is defined as the border that has to be added to each of the two objects to make it cover to other object. A contracting IFS will develop object $x = T_0(o)$ as closely to the attractor $y = o$ as desired. Alternatively, for example, the Hausdorff distance can be used to measure the similarity between IFS and attractor based on the outline. If the object content should also be considered, the template metric or some other measure discussed in the last chapter can be used. One last thought in this respect is using the *fractal dimension* as a description of the self-similarity of an object. For IFS it is defined as follows.

$$f \text{ with } \sum_i c_i^f = 1 \tag{25.3}$$

That is, we are looking for the dimension f that transforms the sum of all contraction parameters c_i associated with contractions T_i to unit size. This fractal dimension f is equivalent to both the *Hausdorff dimension* and the *box counting dimension*.

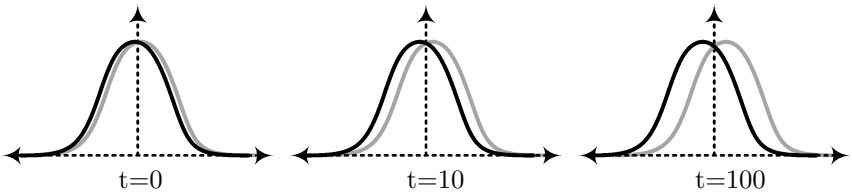


Figure 25.4: Iterative Feature Separation.

The last idea that we would like to present in this section is the *usage of random numbers as arbitrary descriptions*. Of course, without proper improvement arbitrary descriptions will hardly be able to express semantic information.

However, if cleverly chosen and improved they might be reasonable media descriptions. The idea of improvement is illustrated for one description element and two semantic categories (black, gray) in Figure 25.4. If there is a small statistical difference between the initially computed random numbers (here, different mean), we can identify a learning process that moves the two classes further apart. In the second part of the book, we encountered a number of techniques for such discrimination: linear discriminant analysis, intelligent kernel functions, etc. These methods and learning-based categorization methods (e.g. boosting) can likewise be used to improve description quality.

For the initialization problem, it turned out that *random step functions* have a natural ability to provide the required small differences. We suggest using the following function for the definition of $i < m$ description elements f_{ij} for $j < n$ media objects o_j.

$$f_{ij} = \frac{ij+1}{mn} random() \tag{25.4}$$

Here, *random* is a function that returns a random number in the interval $[0, 1]$. Independent of the class labels associated with the objects o_j these arbitrary descriptions tend to have a topology that allows for iterative improvement.

Why should it be interesting to use arbitrary descriptions for semantic learning? There are two general advantages. Firstly, random features can be computed rapidly without considering the media content, i.e. they can be computed in advance. Their only property has to be that they discriminate in arbitrary groupings. Quantitative examples showed that the stated random step function provides this functionality. Secondly, iterative learning moves the semantics problem away from the feature extraction where it can hardly be fulfilled. Improving cleverly chosen random numbers is often simpler and more effective than improving redundant content-based descriptions. In the event, the choice depends on the *quality of the ground truth*. For representative ground truth, arbitrary descriptions are an option. For bad ground truth, they will fail more spectacularly than content-based features.

In this section, we have introduced a number of very simple description methods that can be enriched semantically by intelligent application. Now, it is time to analyze such semantic applications.

25.3 Semantics in Audio, Biosignals and Text

The two remaining sections focus on semantic implementations of the big picture. For selected application types we explain the currently best solution. Like in the last chapter, in this section we focus on one-dimensional data while the next section covers all visual topics. The reason is obvious. Rich semantics

do exist in visual data: object information (signs), their relationships, motion, etc. In comparison, fewer semantic dimensions do exist in one-dimensional data: language in speech, motifs in music, and a few more. Furthermore, audio information is also an important clue in video semantics. We require audio concepts for the recognition of advanced multimodal concepts such as emotions. Hence, we describe this data type first.

After the discussion of semantic audio applications, we introduce one example for biosignals and one for semantic text understanding. Bioinformation and stock data are not considered in this context, because in both cases the semantics go hardly beyond the already discussed methods. For example, the decisive goals of bioinformation detection were already discussed in the first part. Sequence alignment for gene recognition is one such semantic application. Technical chart analysis is performed in the way described in the feature extraction chapters and uses the semantic concepts that were introduced in the last chapter. Any further processing of this data (e.g. taxonomic conclusions, market decisions) is out of the scope of media understanding.

Before we begin with semantic audio applications, however, we would like to emphasize that what we call *semantic* here is clearly inferior to *semi-automatic media understanding*, i.e. *putting the human in the loop* of media understanding. The idea of semi-automatic media understanding is to leave the semantic labeling of objects, patterns, groups, etc. to the user. For example, a visual surveillance application that lets the user label events (e.g. as dangerous/harmless) is a semi-automatic media understanding application. Typical applications are object labeling in image understanding and marking of objects in motion tracking. Recent results of international contests (e.g. TRECVID [279]) show that semi-automatic methods perform two to ten times better than fully automatic media understanding. That is, however semantic our applications are, they are still clearly inferior to the human understanding of semantics.

For semantic audio understanding we would like to discuss two applications: speech recognition and music genre classification. *Speech recognition* is the classic of audio understanding. The semantics in speech and language are obvious and of paramount importance for numerous applications. Early applications included automatic phone answering services and speech-based computer interfaces. The quality of automatic speech recognition has reached a very high standard – in particular for the English and the Spanish language. After some training, recognition rates beyond 99.5% are state-of-the-art.

Figure 25.5 illustrates the speech recognition process. It follows the big picture of media understanding. Speech recognition is a two-step process of training and application. In the training step, MFCC descriptions (see Chapter 13) are extracted for short windows of time (for example, 40ms). These descriptions and ground truth data are used to train hidden Markov models: one per word. The actual categorization process computes the MFCC values for the input sig-

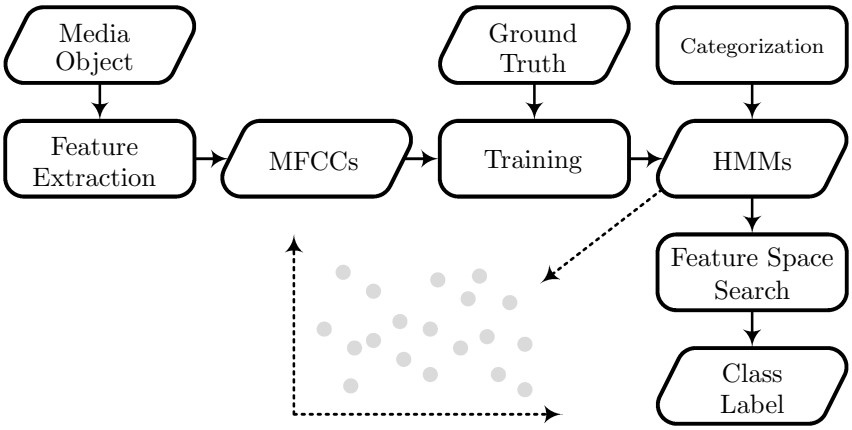


Figure 25.5: Speech Recognition Process.

nal and identifies the most likely hidden Markov model for the input sequence. This evaluation problem is – on the micro level – typically solved by the *forward algorithm*.

However, the actual problem of speech recognition lies on the macro level of categorization. One model per word means, for example, for western languages to evaluate 250 thousand and more hidden Markov model per MFCC sequence. Obviously, this process requires optimization in order to be solvable in (near) real time. Various options exist for the quick traversal of the model space (depicted in two dimensions in Figure 25.5). One obvious approach would be indexing by some tree structure. Another the computation of hash values for models. This approach points in the direction of iterative media understanding. The confusion matrices of the hidden Markov models can be employed for a further iteration of feature extraction and categorization. The process should be optimized in a way that allows quick retrieval of the most likely matches for a new input signal. We conclude that speech recognition is a media understanding of media understanding application that employs spectral psychoacoustic feature transformations and Bayesian categorization methods.

Music genre classification aims at detecting the style of music for some input signal. Genres include classic, pop, reggae, jazz, blues and many more. In contrast to speech recognition, where the central problem is fast search space traversal, the major problem of genre classification is that genres have no clear (fuzzy) definition. In media understanding terms: the descriptions of genres are spread over large parts of feature space and disconnected. Hence, state-of-the-art approaches use categorization methods that produce a summary/view of feature space: cluster analysis, multi-dimensional scaling and the self-organizing

map are typical classifiers. Descriptions are usually extracted using autocorrelation: linear predictive coding and perceptual linear prediction are typical feature transformations. Music genre classification is still in an early stage. The diversity of the genres may be seen as a semantic aspect, but as well as unintended chaos. We believe that significant advances in this domain would require the rigorous definition of genres first.

In the biosignal domain, one very important application is *EEG-based spelling*. The idea of this application is to help people with the locked-in-syndrome to express themselves through the computer. Input stimuli are converted to letters using EEG signals. A typical application scenario consists of the following steps.

1. A sequence of stimuli (e.g. images) is presented: one per letter. The stimuli are presented in an endless loop at a frequency of approximately $\frac{1}{4}$Hz.

2. The user focusses mentally on the letter/stimulus he would like to express. If this stimulus appears, it will cause a P300 event – a strong EEG peak 300ms after the stimulus.

3. The system detects the P300 event and displays the associated letter.

The third step describes the actual media understanding application. Typical feature transformations are short-time energy and peak detection by autocorrelation. Categorization is usually performed threshold-based by simple decision rules.

Early experiments in EEG-based spelling showed that the method is highly reliable. Test users could write at an approximate speed of eight letters per minute for several minutes before they got tired. If the system uses a predictive text application, this equals up to four words.

Our last example is an emerging field in text understanding: *stylometry*. That is the automatic description and categorization of text documents. Applications include quality assessment and the recognition of authorship. In particular, the latter application has received increasing attention in recent years. Stylometry is usually a straightforward implementation of the big picture of media understanding. The employed feature transformations are usually based on text windows of 50-100 words and use averaging and peak detection methods. One example is the counting of the average and frequency of functional words (e.g. relative pronouns), another the counting of seldom pairs of words. The general approach is building a word histogram over each text window, averaging by statistical moments and comparing histograms by distance measures. More advanced categorization approaches make use of neural networks and optimization techniques such as genetic algorithms. However, the categorization in stylometric applications is not different from the categorization in any other

media understanding domain. For the forensic application, the major problem is guaranteeing reliability or, at least, giving a belief score for the stylometric results. Since the style of most authors changes over time, it is hard to provide reliable judgments. Still, we believe that with the increasing availability of copyrighted but easily reproducible digital text, stylometry will become an important frontier of active text understanding research. In this endeavor, the edit history of the Wikipedia will probably become a valuable source of ground truth.

The introduced applications show that tailor-made applications suffer mainly from missing or low quality ground truth and from the over-narrowing problem. Their common advantage is their practical relevance. The presented semantic applications are of immediate use for real people. This relevance justifies the large research effort in these areas.

25.4 Visual Semantic Applications

The final section investigates semantic visual media understanding applications that are of practical relevance. As already stated, numerous semantic concepts are based on the visual sense. The importance of semiotics – the endeavor to develop a language-like symbol system for vision – supports this view. Among the many concepts that are perceived visually, the human face has arguably the highest rank. Cognitive scientists could already discover various neural trails that are activated whenever a human face appears in the visual field. Therefore, we start our investigation with the state-of-the-art in human face recognition. Then, we investigate two applications that are multimodal (audio and vision) but where the visual sense is usually judged prevalent. The first deals with a relational topic: scene grouping in film. The second deals with emotion recognition from video material.

Figure 25.6 sketches the flow in the *face recognition* approach developed by Viola and Jones [383] that may be considered (at least close to) the state-of-the-art in this active research direction. Obviously, face recognition is an important functionality in many media understanding applications. If man is the measure, his face must necessarily be highly relevant. That is equally the case in film analysis as in video surveillance, biometry, content-based image retrieval and other areas.

The Viola-Jones approach is distinguished by its descriptions and the optimization of the categorization approach. The entire process is based on so-called two-, three-, and four-rectangles. Figure 25.7 shows examples. The upper left two-rectangle is computed by summing the brightness values in the white area and subtracting the brightness sum in the gray area from this value. Three- and four-rectangles are computed analogously. These rectangle descriptions are

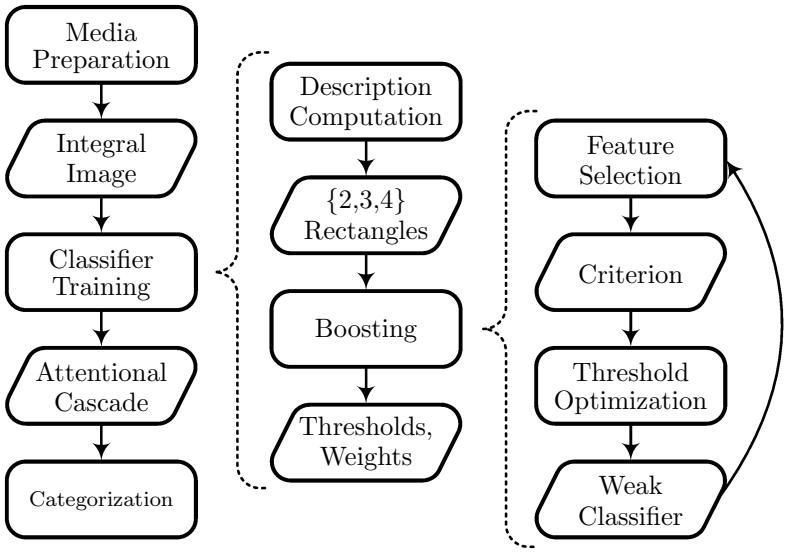


Figure 25.6: Viola Jones Face Recognition Process.

employed for categorization. Three aspects of this approach are noteworthy.

1. The set of description elements is over-complete. That is, the computed description is bigger than the matrix of luminance values if each pixel is considered a region (that is the case in the Viola Jones approach). Hence, there is no information filtering in the feature extraction step, rather the opposite.

2. The computation of the rectangle features is not as resource-consuming as it appears on first sight. The right part of Figure 25.7 shows four so-called *integral images*. The authors of [383] define an integral image as the sum

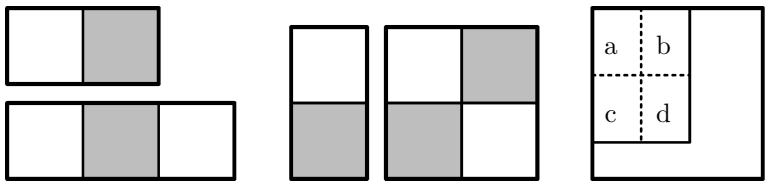


Figure 25.7: Rectangle Features based on Integral Images.

of brightness values in the rectangular region from the upper left corner to any other point. Computing all integral images of an input media object in advance, allows to compute rectangle features quickly by checkers-style addition and subtraction of regions.

3. Rectangle features come very close to arbitrary descriptions. The influence of one brightness value on a number of description elements is – compared to other feature transformations – very big. Therefore, the noise in the input signal is even amplified. The result is a large feature space in which most description elements are junk but some are – hopefully – highly expressive.

The task of the classifier is to identify the expressive description elements. It is, therefore, not surprising that Viola and Jones suggest the AdaBoost algorithm, which is indeed able to identify such description elements quickly and to employ them in simple yet effective decision rules. AdaBoost is used in a meta-process for face recognition. The *attentional cascade* stands for a processing queue in which first a coarse classifier uses coarse two-rectangle descriptions to sort out the majority of false positives, a second classifier works on a finer level (three-rectangles) and the last on four-rectangles. Eventually, a hit for some input face image is identified – hopefully without eliminating the best match in earlier processing steps.

The Viola-Jones approach is highly effective. Following the idea of structural risk minimization, it does not only optimize classification performance (minimization of losses) but at the same time the computational performance. Both the feature extraction procedure and the categorization algorithm are well-suited for algorithmic optimization. The danger of overfitting in the classifier is reduced by the extremely large feature space that introduces a – desired – hardly handleable degree of variance in the feature selection process.

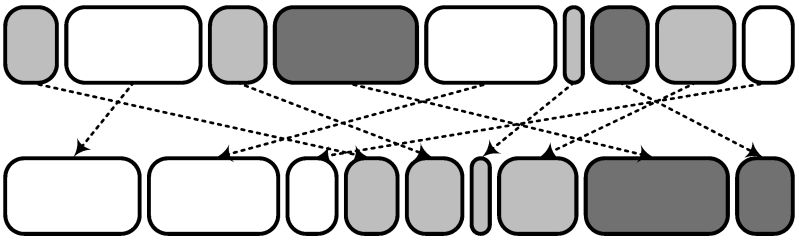


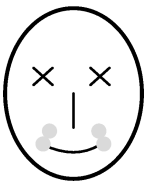
Figure 25.8: Scene Grouping Example.

Scene classification is a central problem of film analysis. The goal is to break up the final cut of a movie, cluster the scenes into shots with the same content,

background and camera parameters and to sequence the scenes in each cluster by their content. Figure 25.8 illustrates this idea. The movie in the top row consists of three threads, for example, someone counting, astronauts in a space ship chatting and the ignition of a space rocket's engine. Scene classification detects the shot boundaries and reorders them by content. The resulting film can be used to investigate aesthetics, semiotic and other clues.

The state-of-the-art in scene classification is to use both audio and visual information to detect shot boundaries in the first step. The temporal segmentation methods introduced in the second part of the book have a hit rate of 99% and more. In the second step, scenes have to be sequenced. For that, descriptions are extracted from the regions next to the boundaries. By crosscorrelation of descriptions of each begin region and all end regions, the most likely matches are identified. Eventually, scenes are sorted by these likelihoods. Descriptions used in this process are color descriptions and local interest points.

The last semantic application that we would like to discuss here is *emotion recognition* from audiovisual content. Emotions are central in many surveillance decision problems and other visual media understanding domains. In video surveillance, it makes a big difference if one person approaching another person shows a smiling face or an angry face. Generally, the visual channel is of higher importance in emotion recognition. However, the auditory information must not be neglected.



Happiness = 6 + 12

Cheek Raiser (6)
Lip Corner Puller (12)

Figure 25.9: What is Happiness?

In the visual domain, again, facial expressions are of highest significance for emotion recognition. Hence, face detection and face recognition are necessary first steps. Then, the facial expression has to be recognized. For that, Ekman has developed the so-called *facial action coding system* (FACS), a standardized set of expressions and a codification of the muscular movement required to form each expression (so-called *action units*). Figure 25.9 shows an example for happiness – the simplest facial expression in the FACS. Happiness is distinguished by raised cheeks and raised lip corners. The FACS distinguishes around 50 action units, different types of head movement, eye movement and gross behavior such as chewing and speech.

The FACS is an ideal foundation for future vision-based emotion recognition. Today, media understanding is not yet able to recognize the action units from

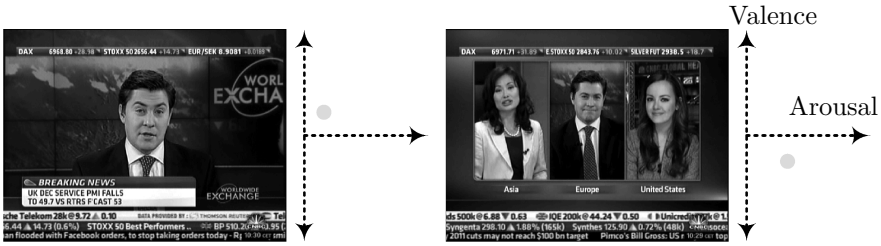


Figure 25.10: Emotion Recognition by Valence and Arousal (© *CNBC*).

motion information correctly. If that was possible, FACS would probably be a reliable system for facially expressed emotions. Meanwhile, emotion recognition is usually based on a dimensional approach, in which emotions are described by their *valence* (positive or negative) and the level of *arousal*. Figure 25.10 shows two examples. Happiness, sadness, anger and many other emotions can be uniquely positioned in this description space.

The media understanding problem in this approach is the transformation of the input media object to the description space. This problem can be solved by an iterative media understanding process. In the first step, state-of-the-art low-level descriptions are extracted. Practically, audio feature transformations such as short-time energy, pitch, MFCC and the logarithmic attack time have proven to be successful description methods. In the visual domain, face moments and those rectangle features that are used for boosting in the Viola-Jones approach have been successful. In the second step, a classifier is used to compute valence and arousal based on given ground truth and all description elements. Here, the k-nearest neighbor approach has proven sufficient for representative ground truth. Eventually, emotions can be derived from valence and arousal by probabilistic inference. So far, the state-of-the-art in emotion recognition has hardly left the diagonal of arbitrariness of the ROC chart. However, this is an active frontier of media understanding research and we are positive that the near future will see significant advance in this field. In particular, making effective use of the FACS would lead to a major step forward in emotion recognition.

The conclusion of this chapter is that semantic applications are an important frontier of media understanding, because eventually, users require applications that work. An application is semantic if it is neither too primitive nor to clever. We introduced the terms *subsemantic* and *supersemantic* to denote these two insufficiencies. In particular the second one has been widely neglected in the past. Semantic applications reduce the semantic gap, no surprise, and also the level of polysemy by focussing on the context of the user. The drawbacks of these often iterative methods are high dimensionality of the feature space and generally high resource consumption. The major problem of the domain is over-

narrowing: the tendency to build applications for a domain too limited to be relevant in practice.

The last two chapters, template matching and semantic applications, were dedicated to practical problems of media understanding, mostly associated with the description step of the big picture. In the next two chapters, we return to more theoretical problems – though important applications do exist: dynamic filtering and the frontiers of learning. As in the first two parts, by these chapters we make the transition from media description to categorization.

Chapter 26

Convergent Filtering

Develops a model of convergence for iterative filtering processes, discusses learning vector quantization, the Kalman filter for scalar quantization and quantization by associative memories such as the Hopfield network and the Boltzmann machine.

26.1 Models of Convergence

This chapter and the next are dedicated to dynamic processes. In this chapter we *focus on information filtering* and introduce methods that provide data quantization by convergence over time. In the next chapter, we investigate the behavior of classifiers over time, that is, their dynamic learning and application behavior. Of the four sections of this chapter, the first deals with theoretic aspects of dynamic filtering: forms of convergence, limits of the filtering process and similarities to related processes, in particular, optimization. The remaining sections introduces concrete models for convergent filtering. The second section discusses vector quantization as a form of convergence towards a given limit. Section 26.3 introduces the Kalman filter as a near-optimal solution for convergent filtering under uncertainty. Eventually, we introduce two pseudo-neural processes: the Hopfield network and the Boltzmann machine, that extract scalar quantization from the topology of the input data space.

Convergent filtering is a true frontier of media understanding research. Some methods, such as the Kalman filter, are well-established for some applications (e.g. point tracking) but neglected in others (e.g. computation of informative moments). Other methods are hardly used today, even though a number of

interesting applications would exist. The goal of this chapter is to introduce and compare important converging filtering methods. We would like to make the reader aware that, here, a class of algorithms exists that have common attributes and several potentials for application. We are positive that the existing approaches could be used beneficially in media understanding and that further convergent filtering approaches that combine features of the existing ones would be interesting to have.

In this section, we focus on the ingredients of the convergent filtering model. Dynamical aspects are – where possible – referred to the next chapter. First, we introduce the general principle of convergence in the form of the cybernetic stability criterion. Then, we discuss several operationalizations, i.e. convergence curves, and their parameters. Eventually, we discuss the convergence process in theory and practice. As we will see, it is a dynamic process with all typical features. The section closes with a brief discussion of the common aspects of quantization and optimization.

Quantization is the central term in this chapter. The feature extraction chapters and, in particular, the analysis of the building blocks of feature transformation showed that quantization steps occur in all relevant description methods. Often, quantization introduces essential world knowledge and provides the foundation for expressive descriptions. Convergent filtering aims at the summarization of input data – over time – in stable descriptions. Hence, convergent filtering can be seen as a form of quantization.

Temporal processes are highly interesting for quantization. Improvement over time allows to approach the optimal value for given data in a trial and error manner, to make use of belief scores and to escape local optima. The risk of temporal filtering, however, is that the approximation process begins to oscillate or even to show chaotic behavior. Obviously, for quantization such behavior has to be avoided. We require a stability criterion for the dynamic filtering process. Cybernetics, the theory of dynamic feedback systems, provides such a criterion in the following form. For a feedback process $F(t)$ given in the form of a differential equation $f'(t) = F(t, f(t))$ the solution $x(t)$ is stable if the following condition holds

$$\big|f(0) - x(0)\big| < \epsilon_0 \wedge \big|f(t) - x(t)\big| < \epsilon_t \tag{26.1}$$

Here, $0 < \epsilon_{t+1} < \epsilon_t$ are limits that enforce the convergence over time $t < \infty$. Please note the similarity of the stability criterion with the contraction condition of iterated function systems (last chapter). The idea is the same: iterated function systems have to be convergent processes. A process is said to be *asymptotically stable* if the stability criterion holds as well as the following condition.

$$\lim_{t\to\infty} \left| f(t) - x(t) \right| = 0 \tag{26.2}$$

That is, the output of the process is normalized to a time average (here, simply the mean) of zero.

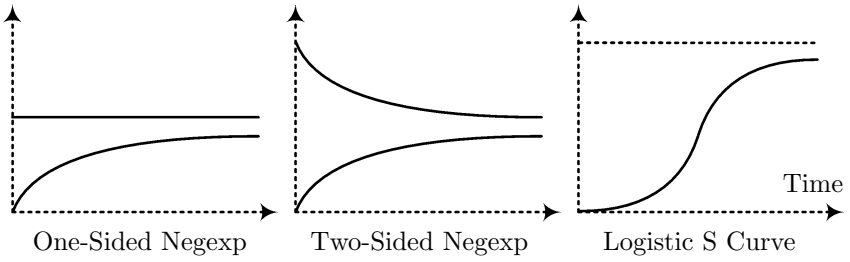


Figure 26.1: Convergence Curves.

Figure 26.1 shows typical convergence curves that fulfill the stability criterion. It has to be remarked that since a dynamic process that is controlled by feedback must necessarily oscillate, a true convergence curve would have to have high-frequency components. However, we are not interested in the local behavior of the curves here, therefore, we focus on the *convex hull* of practical convergence curves. The leftmost curve shows a typical one-sided negative exponential curve. The filtering process that implements this behavior is highly efficient. After relatively short time (a high *rate of convergence*) it provides already a fair estimate of the result. The two-sided negative exponential curve approaches the *point of stability* from both sides, hence, having higher oscillation and smaller belief in the first iterations. This is arguably the most typical convergence process. For example, the Kalman filter follows this characteristic. The efficiency of the process depends on the degree of over-linearity of the curve.

The rightmost element of the figure shows the most common *sigmoid curve*, the *logistic S curve*. This curve is often used to describe learning processes that consist of an initial stage (exponential growth) and a saturation stage in which the domain is mostly processed (learning is complete). The curve is defined as follows.

$$f(t) = \frac{1}{1 + e^{-t}} \tag{26.3}$$

The logistic convergence criterion will usually be one-sided. The upper limit is given by the size of the set/space on which the underlying process operates (e.g. a feature space). A practical example of a filtering process that will follow the logistic S curve is learning vector quantization, discussed in the next section.

Remark: Time is the essential parameter of convergence curves. The treatment of time is an important aspect of dynamic filtering processes. Most processes use a discrete time model in which the system changes as a whole from step to step. Within one step, all data are held constant. This model is, for example, employed in the Kalman filter and in associative memories. Learning vector quantization may be seen as an exception as it handles only one input value and, hence, only a small fraction of the model in one time step. In Chapter 29 we ill introduce dynamic models that work with continuous time and do not hold the model constant.

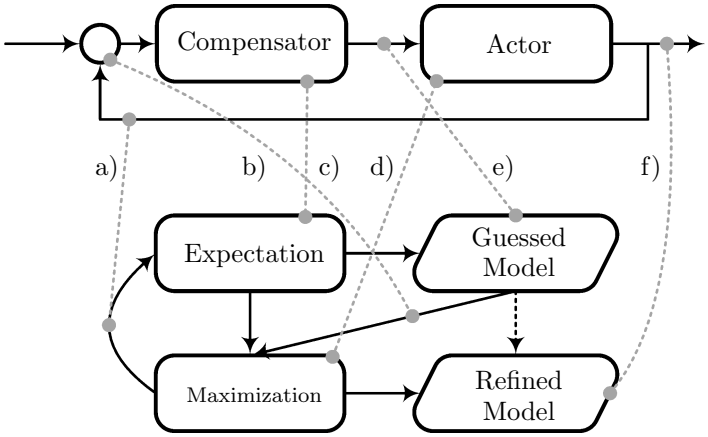


Figure 26.2: Convergent Filtering Model and Example.

Convergent filtering is a dynamic process, we said. Hence, the filtering model will be derived from the general model of dynamical systems. Figure 26.2 provides an example in the form of the standard feedback system. Since dynamical systems require control by feedback, we consider this model a fair choice for the convergent filtering model. Below the feedback model, the figure shows the *expectation maximization* approach as one example for a dynamic filtering process. Following the two-sided negative exponential convergence curve, expectation maximization usually has a high rate of convergence. For example, the Kalman filter uses an expectation maximization process. The gray lines connect components of the theoretical model with components of the practical example. As we can see, every element of the implementation exists also in the model and vice versa. This underlines the fact that the feedback model is a good choice for the general convergence filtering model. It consists of the elements listed in Table 26.1.

The actor provides the actual estimate. The estimation process is based on the model (e.g. a form of belief), which is provided and maintained by

Element	*Name*	*Type*	*Example*
a	Feedback	Data	Estimate
b	Integrator	Process	Add Operator
c	Compensator	Process	Density Estimation
d	Actor	Process	Extrapolation
e	Model	Data	Probabilistic Density
f	Output	Data	Final Estimate

Table 26.1: Elements of the Convergent Filtering Model.

the compensator. The work of the compensator is based on past estimates (feedback), integrated by some add function. The final output is the stable result. The duration of the filtering process is primarily determined by the behavior of the compensator. The better the model, the higher the rate of convergence. The characteristics of the convergence curve are determined by the actor, since this component makes use of the model and produces the actual estimate.

We would like to close this section with a brief discussion of the communalities and differences of quantization and optimization. Above, we argued that quantization over time is a dynamic process that has to converge towards some stable point. The same description can be given for optimization processes, only that we require the stable point additionally to be the best with respect to a given goal. Please note that this is not expected from a quantization process. We will hardly ever have an understanding of optimality in quantization.

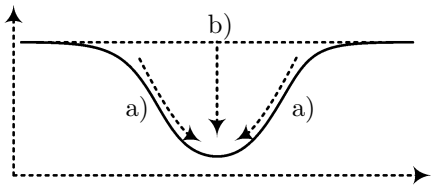


Figure 26.3: Optimization (a) and Quantization (b).

Figure 26.3 illustrates the communalities and differences of quantization and optimization processes. Both processes have in common that they search for a point representation of a given space. The space will in both cases be non-linear. Hence, the search problem will not be trivial. In the example, the optimization process will search for the minimum of the curve. This can be approached by linear search, hill climbing or one of the algorithms introduced in the second part of the book. Eventually, the optimization algorithm should end up at the

minimum. In contrast, the quantization process will chose its representation not only from the given data but also from constraints in the model. For example, it will not allow the selection process to exceed beyond a pre-defined limit. Hence, quantization can be seen as a form of local optimization in which global aspects of the input data are only considered in so far as every component (often, of fixed size) of the global signal is represented by one local quantization. If optimization is as flexible as a worker, quantization is like a crane.

In summary, in this chapter we try to solve the information filtering problem (in particular, quantization) by dynamic processes. In order to be useful, these processes have to be stable in the sense that they follow a reasonable convergence curve. The essential building blocks of the dynamic filtering model are the same as in general feedback systems. The central element is the model data, often, a probability distribution/set of belief scores. The filtering process employs the model for improving an initial estimate until it reaches a stable point. In the three remaining sections of this chapter we will see that these building blocks can be used to categorize components of practically relevant algorithms successfully.

26.2 Vector Quantization

Quantization is ubiquitous in media understanding. The central idea of quantization is the coarse representation of input values, or, to provide a mapping from a larger set onto a smaller set that preserves the relationships of the original data. A typical quantization function is rounding. In the feature extraction chapters, we have seen that quantization is of highest significance. It is used, for example, for psychophysical modeling (often, the logarithm) or for normalization (e.g. in interest point descriptions). This section focusses on a particular form of quantization: *vector quantization* by iterative processes. The central algorithm is *learning vector quantization*. In addition, we take the opportunity to introduce a few relevant terms in the context of quantization and to review others that were introduced earlier.

Generally, two types of quantization methods can be differentiated: *scalar quantization* and *vector quantization*. The first method tries to optimize the representation of one value while the other performs quantization for entire data sets. The rounding function mentioned above is a typical scalar quantization method. The k-means algorithm is a vector quantization method. Both approaches have in common that they map many values onto few – often, uncountable many to countable few. A major difference is that the rounding function does not require external data (e.g. world information): rounding can be performed without an understanding of the output space. In contrast, the k-means algorithm requires *references* that define the topology of the output space. This differentiation of quantization methods is crucial for their understanding. Some

methods are *goal-centered* while others are *self-centered*. In the first case, the quantization result can be controlled by parameters, in the second case not (or, hardly). Methods that cannot do without external information are, for example, the psychophysical transforms already mentioned and ground truth-based methods.

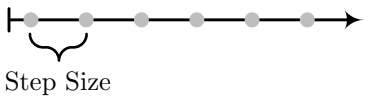


Figure 26.4: Granularity of Quantization.

The effect of quantization can be measured by two indicators.

- Granularity
- Quantization error

Granularity, as shown in Figure 26.4, is an indicator only for self-centered methods, for goal-centered methods it is a parameter, then often referred to as *step size* (e.g. the step size in audio feature transformations). The quantization error was already introduced in Chapter 19. Generally, it measures the rounding error of the quantization function.

$$q = \sum_i |\bar{f}_i - f_i| \tag{26.4}$$

As the formula shows, the quantization error is simply the city block distance from the output $\bar{f}$ to the input f of quantization. Hence, this measure is applicable to self-centered and to goal-centered quantization algorithms. Of course, in the latter case the quantization error will be influenced strongly by the layout of the references.

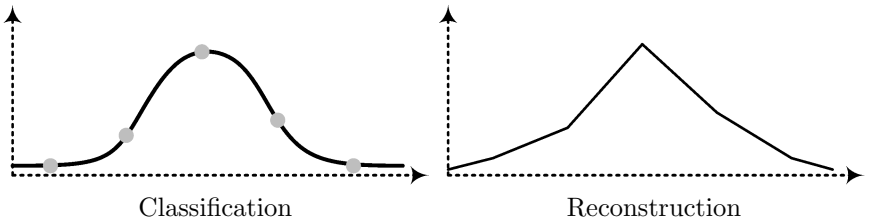


Figure 26.5: Quantization is a Two-Step Process.

Furthermore, quantization methods are often segmented into two steps.

1. Classification
2. Reconstruction

Figure 26.5 shows an example. In the classification step, the input signal is reduced to a few points that express the characteristics of interest. In the figure, the bell curve (input) is quantized to a few points (gray). In goal-centered methods, the orientation points of classification are given. For self-centered methods, they are usually distributed according to some distribution (e.g. uniformly). The reconstruction step extends the relevance of the orientation points onto the entire input space. The result is a coarse representation of the original input signal. The amount of coarseness is determined by the degree of violation of the Nyquist law. The level of coarseness can be expressed by the quantization error.

The k-means algorithm is a good example for the classification-reconstruction scheme. In the classification step, input vectors are associated with their nearest references. In the reconstruction step, a Voronoi tessellation is computed from the reference vectors. The result is a quantization scheme for vectors that spans over the entire feature space.

Vector quantization differs in one essential point from scalar quantization: The amount of quantization varies between vector elements. The global quantization is determined by some similarity or distance measure. Every vector element contributes to the global quantization but the amount of contribution depends on the distribution of the values of the elements. For example, a constant vector element will contribute little to coarse representation. A highly variable element will contribute strongly. The general strategy of vector quantization is the same as *competitive learning* in neural network theory: *the winner takes it all*. Vector quantization is generally goal-centered, i.e. the desired topology has to be given in the form of references. Then, for each input vector the *winning node* is detected as the most similar reference. Like for k-means and the self-organizing map, Minkowski distances are the preferred (inverse) similarity measures. Optionally, the winning node is moved a little bit in the direction of the input vector. The *little bit* is determined by a *learning rate*. Hence, competitive learning has two determining parameters: the references and the learning rate.

The most important implementation of vector quantization for media understanding is the *learning vector quantization* algorithm proposed by Kohonen. It implements exactly the competitive learning algorithm (see Figure 26.6). The resulting references (codebook vectors) are used instead of the input data as quantized representatives. Hence, this algorithm is highly similar to k-means categorization and the self-organizing map (SOM). Compared to the first, it adds a learning step that adapts the references (i.e. an integrator and feedback).

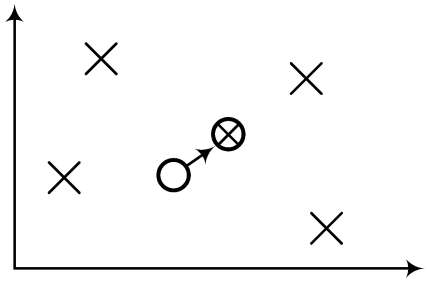


Figure 26.6: Learning Vector Quantization Principle.

Unlike the SOM, learning vector quantization does not consider the neighborhood of the winning node. Only this reference is adapted, not the others. The algorithm is therefore somewhere between k-means and SOM. The resulting categorization is less smooth than a SOM categorization. This may be the reason why the SOM is considered a classifier and learning vector quantization is seen as a vector quantization function.

Learning vector quantization is a highly flexible method. Convergence follows the one-sided negative exponential model. That is, the initial direction of learning is pursued until the optimum is reached. The learning process allows to correct badly chosen references. It is therefore usable for text quantization (e.g. normalization of n-grams) as it is applicable for improving the discrimination ability of audio descriptions for music genre classification. The important aspect of vector quantization is that it is data-driven. The implementing algorithms employ the references as their model and transform the input data accordingly. The trust in the model is beyond question. In the next section, we introduce a filter that takes the uncertainty of the model into account.

26.3 The Kalman Filter

The *Kalman filter* [184] is the state-of-the-art in the estimation of the state of a noisy and/or uncertain process. In this section, we introduce a practical form of the Kalman filter that can be employed for convergent filtering. In the description, we avoid going to deep into the theory of the filter. Rather, we present practical media understanding applications for Kalman filtering. In fact, the Kalman filter is a convergent filter par excellence. First, we introduce the model of the process that is being approximated by the filter. Then, we discuss the convergent filtering process and the model of the filter. Eventually, we sketch a few practical applications.

The Kalman filter describes a noisy process of the following form.

$$x_t = F(x_{t-1}) + N(0, \sigma_n) \tag{26.5}$$

The output x_t at time t is produced by a process F that is based on the last state. The Gaussian noise component $N(0, \sigma_n)$ influences every iteration of the process differently. This process is very similar to a Markov process of first order. In fact, the Kalman filter can be used to approximate the parameters of a Markov process. Moreover, the entire Kalman filtering process is isomorph to the expectation maximization algorithm typically employed to model the confusion matrices of Markov processes. In the second part, we mentioned that Gaussian mixture models are frequently used for initializing hidden Markov models. This is an application example of this process.

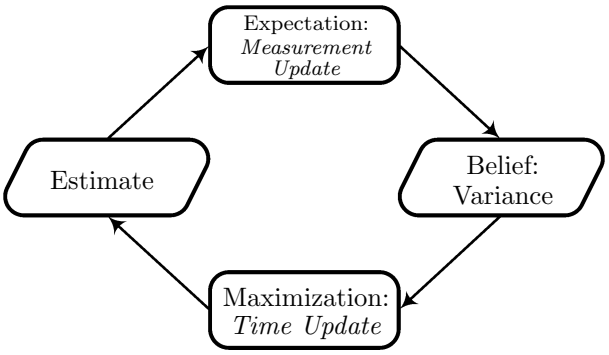


Figure 26.7: Kalman Filtering Process.

The Kalman filtering process is illustrated in Figure 26.7. The two steps of the process are iterated over time. In the expectation step, the measurement is updated, i.e. the model that is used to estimate the output of the filter is adapted by the most recent measurement. In the maximization step, the refined filtering model is used to adjust the estimate of the process F. Since the process output is determined temporally, this is called the time update.

More precisely, the initialization of the filtering process is performed as follows.

$$f_0 = x_0 \tag{26.6}$$

$$f_1 = \mu(f_0, x_1) \tag{26.7}$$

That is, the process estimate f_1 at time 1 is just the mean of the first two measurements $x_{\{0,1\}}$. Based on this initialization, the first measurement update is performed as follows.

$$b_t = \frac{\sigma_{t-1}}{\sigma_t + \sigma t - 1} \tag{26.8}$$

Here, b_t is the belief in the measurement at time t. If the standard deviation of the data stream generated by the process is increased by measurement x_t, then the belief term moves towards zero. If the standard deviation becomes smaller, the belief moves towards one (trust). Formally:

$$\sigma_t \rightarrow \max \Rightarrow b_t \rightarrow 0 \tag{26.9}$$

$$\sigma_t \rightarrow 0 \Rightarrow b_t \rightarrow 1 \tag{26.10}$$

The belief b_t is the model of this simple form of the Kalman filter. It is used in the time update in the following way.

$$f_t = f_{t-1} + b_t\big(x_t - f_{t-1}\big) \tag{26.11}$$

The new estimate depends on the old estimate (output), the belief in the measurement (feedback, compensator) and the difference of the new measurement and the old estimate (integrator). The latter term is the time update. A large time update means that the old estimate did not predict the next output well. Hence, the belief in the model will fall. If the time update is small, the belief in the model will rise. In consequence, the product of belief model and time update will be maximal for medium time updates. Only such updates will make the new estimate significantly different from the old one. If we denote the time update as $T = x_t - f_{t-1}$, we can write this relationship as follows.

$$\sigma_t \rightarrow \max \Rightarrow b_t \rightarrow 0 \wedge T \rightarrow \max \Rightarrow b_t T \rightarrow 0 \tag{26.12}$$

$$\sigma_t \rightarrow 0 \Rightarrow b_t \rightarrow 1 \wedge T \rightarrow 0 \Rightarrow b_t T \rightarrow 0 \tag{26.13}$$

For high variances in the output of process F we trust in the estimate. For low variances there is no need to refine the estimate. The power of the Kalman filter lies in medium-sized changes of the measurements. Then, it computes an estimate that predicts the output well and eliminates the noise component.

Before we discuss media understanding applications of the Kalman filter we would like to compare it to statistical moments and interestingness measures. If we ignore the temporal aspect of the approximated process – which is anyway always the case in media understanding applications – the Kalman filter can be interpreted as an algorithm that computes a first-order moment of the input data in a similar fashion as the mean shift algorithm. Hence, it appears reasonable to investigate whether or not the Kalman estimate is correlated to mean, median or interestingness measures such as information entropy. We investigated theses

measures for random data sets (uniformly distributed and normally distributed). Quantitative analysis did not show any significant correlations between these moments. In detail, the only similarity that could be identified was between entropy and variance, which underlines the characteristics of interestingness measures as moments of second order. Among the moments of first order, it was astonishing to see that – even for short random sequences – the mean is significantly different from the Kalman estimate. Therefore, we consider it advisable to use both procedures to approximate data in media descriptions and select the better one by ground truth-based evaluation or factor analysis.

A second point of interest is the influence of the order of the input data on the Kalman estimate. Generally, the filter shows two-sided convergence with the characteristics of the negative exponential curve. Quantitative analysis showed that the effect of reordering depends on the variance in the data. If variances are generally small, reordering has hardly an effect on the Kalman estimate. If variances are high, nonsurprisingly, the estimates become significantly different. We conclude that the applicability of the Kalman filter for quantization in media understanding depends on the degree of variation in the media objects and descriptions. If the number of degrees of freedom and their scale is limited, Kalman filtering may be an interesting alternative to mean filtering.

Figure 26.8: Kalmanface Example (© *CNBC*).

One such application is illustrated in Figure 26.8. We employed the Kalman filter successfully for face averaging in the context of face recognition. The approach is intended as a preprocessing step. It takes the n input images (left part of the figure), normalizes them to the same size and – if necessary – shifts the center (e.g. to the nose tip) and rotates the view. Then, an average face image is computed by Kalman filtering for each sample location. The resulting face estimate (a so-called *Kalmanface*) is illustrated in the center element of Figure 26.8. The Kalmanface is a mixture of the original face information and the belief in the various features. The statistical properties of the Kalmanface are significantly different than of the mean image. The overall entropy is higher. Eventually, the Kalmanface can be converted into a description by coarse representation (quantization) and, for example, a zigzag scan. The resulting description can be used

for face recognition directly or as input for some meta-algorithm.

Typical applications of the Kalman filter include smoothing of tracking data, forms of regression and extrapolation. Hence, the filter is tailor-made for the analysis of stock data. Surprisingly, it is hardly employed in this domain. One interesting application would be the averaging of stock data segments, for which a sliding average is frequently used. Another application would be the prediction of the missing part of a chart (e.g. a cup with holder) based on earlier measurements. Eventually, in the audio domain, the Kalman filter could be used for the estimation of short-time energy – similar to the mean, etc.

In summary, the Kalman filter estimates (temporal) processes. It computes an optimal estimate under uncertainty, for example, introduced by noise. Applications include information filtering but as well quantization in feature transformations and categorization algorithms. The Kalman filter is due to its simplicity and the intelligent combination of belief and empirical measurements of highest practical importance. In media understanding, it can be used on descriptions and directly on the media data.

26.4 Associative Memories

In the final section of this chapter we move from belief-based filtering on to temporal feedback models. *Associative memories* are often modeled as neural networks. Like the human cognitive system, associative memories search for a stable state, i.e. a *conflict-free system*. The optimization of the individual neuron (for example, standing for one scalar value) is influenced by all other values and influences the optimization of all other values. Associative memories are used for vector quantization.

But associative memories achieve more. The central ability expressed by the adjective *associative* (or, *auto-associative*) is to retrieve one part of a data vector on the presentation of another part. The two models discussed in this section, the Hopfield network and the Boltzmann machine are both capable of auto-association. Auto-association is an important ability in text retrieval. It can, for example, help to solve the co-reference problem (for example, replacing the subjective by a personal pronoun).

Remark: Learning vector quantization – in fact, every form of vector quantization – will also be able to show associative behavior, if an appropriate distance/similarity measure is chosen for the selection of the winning node. For example, the application of predicate-based measures such as the Hamming distance (P3 in Appendix B.2) on binary vectors creates a vector quantization method that shows significant similarities to a Hopfield network.

In the remainder of this section we describe the Hopfield network first, then the Boltzmann machine and, eventually, we discuss applications of both. As we

will see, both approaches share a large number of system properties.

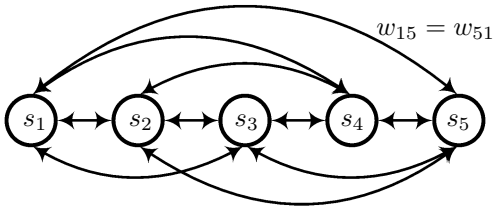


Figure 26.9: Hopfield Network Architecture.

Figure 26.9 shows the design of the Hopfield network. The layout is isomorph to a Markov process of first order. That is, every state s_i is connected to all other states. The connections are weighted – the Hopfield network is a neural network –, and the weights are symmetric. That is, $w_{ij} = w_{ji}$ for the connection of states s_i, s_j. Every state $s_i \in \{-1, 1\}$, hence the state vector is a binary vector (-1 standing for zero). Such a network is frequently called a *recurrent network*.

The firing rule of the Hopfield network is the same as the one of the standard *McCulloch-Pitts neuron*. The state of the receiving neuron is determined as follows.

$$s_i = sgn\Big(\sum_{j \neq i} w_{ij} s_j - \epsilon\Big) \tag{26.14}$$

Here, ϵ is the firing threshold of state/neuron s_i. The actual application is performed in the following steps.

1. Set all states to the values of the elements of the binary input vector (Element f of Table 26.1).
2. Compute the new value for each state according to the firing rule. It is important to note that the Hopfield network uses a *pulsed time model*. Every new state is computed from the entire old state vector (Elements b, e of the table).
3. Update all states simultaneously (Elements c, d of the table).
4. If the difference between the old state vector and the new one exceeds a predefined threshold, return to the second step (Element a of the table).

The Hopfield network is a pulsed iterative process that computes the stable equivalent for each input vector. If the input vector is only given partially, the missing values are (partially) reconstructed. The application algorithm has two-sided negative exponential convergence characteristics. Convergence is reached quickly. The number of necessary steps rises with the number of states.

The initialization procedure of the Hopfield network employs the principle of the *generalized Hebb rule*. After the number of states has been defined (e.g. by the dimensionality of a feature space) the weights of connections are set as follow.

$$w_{ij} = w_{ji} = \frac{1}{n} \sum_{x=1}^{n} f_{ix} f_{jx} \tag{26.15}$$

Interestingly, there is no learning involved in the initialization. The $x < n$ training vectors f_{ix} with states/elements iterator i are employed in the form of the outer product. Hence, the weight of a connection is determined by the *average perfect similarity* of the involved states/description elements. Redundant description elements will have high weights while variant/descriminative elements will receive low weights. The Hopfield network may therefore justly be called a quantization method. It reduces the discrimination power of the input data.

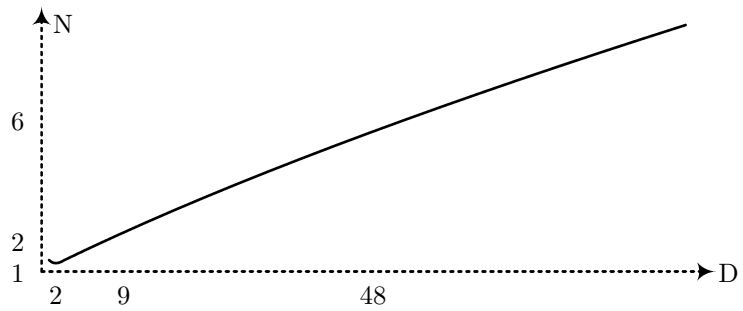


Figure 26.10: Hopfield Network Capacity.

The configuration of a Hopfield network can be expressed by two indicators. The *average weight* is just the mean over all connection weights. This indicator expresses if the network converges quickly (high value) or slowly (low value). The *capacity* N of a Hopfield network with D states is approximated by the following relation.

$$N < \frac{D}{2 \log D} \tag{26.16}$$

In 99% of all cases the number of patterns (codebook vectors) that can be remembered by the Hopfield network will not exceed N. Some examples from Figure 26.10: For two patterns we require nine states, for six patterns $\sim$ 48 states. For higher values, the curve is approximately linear.

In summary, the Hopfield network implements a form of gradient ascent towards attractors that summarize the input patterns. Training is straightforward and quick. The fundamental problem of this network is the dependency on the training data. The initialization step leaves no room for the correction of outliers and noise.

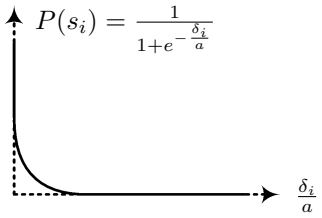


Figure 26.11: Boltzmann Cooling Scheme.

The *Boltzmann machine* is in idea, design an application very similar to the Hopfield network. The major differences are:

- States are defined as $\{0, 1\}$.
- Some connections do not exist, i.e. $w_{ij} = w_{ji} = 0$.
- Not all states are adapted in each iteration of the application process.
- Training is based on an iterative learning process.

The latter two differences require detailed explanation. In the application process, states (and the weights connected to them) are selected stochastically for adaptation. The probability of selection of one particular state is defined as follows.

$$P(s_i) = \frac{1}{1 + e^{-\frac{\delta_i}{a}}} \tag{26.17}$$

Figure 26.11 illustrates the characteristics of this function. The parameter δ_i is the firing state of the neuron. Like for the Hopfield network it is defined in analogy to the standard neuron.

$$\delta_i = \sum_{j \neq i} w_{ij} s_j - \epsilon \tag{26.18}$$

The probability of selection increases with a and decreases with δ. States are selected by maximum likelihood. A Boltzmann machine that selects only the most probable state in one iteration is called *sequential*, any other form *parallel*.

The parameter a stands – like in simulated annealing – for the temperature of the application process. The higher a, the more frequently states are adapted. Lowering a freezes the solution represented by the state vector (*thermal equilibrium*). The states of a Boltzmann machine behave like the particles of the *Boltzmann distribution* – hence the name.

The Boltzmann machine employs an iterative learning algorithm that is similar to cross validation. Initially, all weights are set to zero. Then, the following scheme is executed.

1. Select a subset of all training vectors (e.g. one) and apply them sequentially on the network. In this process, collect statistics about the activation of the weights.

$$P(w_{ij}) = \frac{\text{Number of times } s_i = 1 \wedge s_j = 1}{\text{Number of times } s_x = 1} \tag{26.19}$$

   Here, s_x stands for any state in the network.

2. Select a random training vector and compute the same statistics during the application process: $\bar{P}(w_{ij})$.

3. Adjust the weights of all connections by the following rule.

$$\bar{w}_{ij} = w_{ij} + a(\bar{P}(w_{ij}) - P(w_{ij})) \tag{26.20}$$

   The new weights $\bar{w}_{ij}$ are influenced by the learning rate a and the precision of the individual application process.

4. Return to the first step until the difference of new and old weights is below a predefined threshold.

The learning algorithm of the Boltzmann machine creates an exponential training effort. Training and application are characterized by two-sided negative exponential convergence. The benefit of the learning algorithm is that the Boltzmann machine is more likely to identify a global optimum than the Hopfield network. However, practically every non-trivial Boltzmann machine will not terminate in reasonable time. Boltzmann machines are therefore only of theoretical use while Hopfield networks are heavily employed for vector quantization and data association.

In this chapter, we have presented a number of algorithms for convergent filtering, mostly quantization, of data vectors. In the domain of scalar quantization, the Kalman filter is of highest significance. For vector quantization, learning vector quantization and Hopfield networks can be used. Both methods

have the characteristics of associative memories. In the first case, the resulting quantization will depend on the chosen similarity measure. The solution of the Hopfield network depends on the training data. In general, convergent filters have a positive influence on the dimensionality problem of media understanding. Despite the filtering effort, the effect on system performance will mostly be positive. Some of the methods are tailor-made for the elimination of noise components. On the other hand, there is a tendency that media semantics will be destroyed by such methods. In conclusion, the selection/definition of a one-fits-all filter for media understanding is a true research frontier. It is generally advisable to use the Kalman filter on scalars in situations of uncertainty. Vector quantization is of use wherever the media understanding problem needs to be simplified without taking the risk of destroying the relationships between description elements.

This chapter introduced methods for *filtering over time*. In the next chapter we move to *learning over time*. We review the limits of learning, of particular learning algorithms, and we try to estimate the risks hidden in dynamic learning algorithms.

Chapter 27

Frontiers of Learning Machines

Reviews communalities of categorization methods, presents a system of learning bounds, introduces fundamental methods of dynamical systems and applies these methods on dynamic classifiers.

27.1 Analysis of Categorization Methods

This chapter gathers a handful of advanced topics on machine learning. The major theme is the investigation of the *behavior of classifiers over time*. For this purpose, we introduce some tools from dynamical systems theory and chaos theory. These are applied on dynamic categorization methods in order to identify potential inconsistencies in their behavior. This chapter is related to the previous one. There, we investigated convergence in selected filtering methods all of which are related to categorization methods. In this chapter, we investigate oscillation and – if existent – strange attractors, i.e. the opposite of convergence. Where appropriate, we refer back to the theory presented in the previous chapter.

This first section serves as an introduction. We pick up the thread of Chapter 11 and investigate communalities and differences of the introduced categorization methods. The second section deals with the limits of categorization. We review the major theories in this area and discuss the current state of affairs at this frontier of machine learning. The last two sections focus on the central topic: dynamical systems theory and its application on potentially interesting dynamic classifiers. We will see that a number of classifiers show oscillating behavior and

some are for specific configurations even suspiciously close to chaotic behavior.

We consider the issues discussed in this chapter a true frontier of media understanding. Categorization is the crucial conversion step from media summaries to semantic categories. In this respect, oscillation and chaotic behavior cannot be ignored. For example, the common practice to abort the classifier training process if it does not converge in acceptable time and simply restart it (e.g. training for Gaussian mixture models) is not a sufficient treatment of what is going on in the training process. We require an understanding of what happens in these dynamic processes in order to make sure that the fundamental intentions of categorization remain preserved.

In the remainder of this section we review the already introduced classifiers. We compare them by the complexity of their models, their training processes and their micro processes. Alongside, we summarize the typical micro processes implemented in the individual classifiers. The knowledge compiled and acquired in this section will provide a sound basis for the advanced concepts discussed in the three following sections.

In Chapter 11, we identified four building blocks of categorization: quantization, similarity judgment, model estimation and learning/refinement. The last two describe the training process and, sometimes, the application process. The first two are mostly implemented in the micro process, i.e. the actual comparison of pairs of stimuli (or: references). The detailed analysis presented in this section deals with both aspects: we discuss properties of the micro process as well as of the training/application process. First, we rank the classifiers by the complexity of their models. Then, we discuss a list of micro processes. These are required for a ranking of the classifiers by the complexity of their micro processes. Eventually, we provide a portfolio of classifiers by model rigidity and model complexity.

Table 27.1 ranks the classifiers introduced in the first two parts of this textbook by the complexity of their models. See Chapter 29 for an explanation of the perceptron and of radial basis functions. Cluster analysis is top-ranked, because it has no model at all. The application process is performed directly on the data and the result is just a representation of this data. Bayesian networks with their large number of required confusion matrices are on the other end of the list. This group of methods (that includes Markov processes) arguably requires the largest models.

The fundamental types of models are densities, references, thresholds and weights. A density function measures the frequency of appearance of a (multi-dimensional) stimulus that belongs to some scale (dimension). Densities are global descriptions of events. In comparison, a reference is just one example for a semantic context. Thresholds contain even less information. The typical threshold is just a limit for one dimension of a description space. Hence, it implicitly contains a reference to the corresponding description element. Eventu-

Classifier	*Size*	*Model Type*	*Description*
Cluster Analysis	1	-	-
Decision Stump	2	Thresholds	One Threshold
Random Selection	2	-	Random Threshold
Boosting	3	Weights	Weights, Thresholds
Decision Tree	3	Thresholds	Rules with Thresholds, Weights
Gaussian Bayes Classifier	3	Densities	Means/Deviations
Support Vector Machine	3	Weights	Hyperplane Configuration, Threshold
Vector Space Model	3	Thresholds	Query, Threshold
K-Means	4	References	Codebook, K
Self-Organizing Map	4	References	Codebook, Learning Rate
Linear Discriminant Analysis	5	Densities	Class Means/Deviations, Labels
K-Nearest Neighbor	6	References	Labeled Samples, K
Mixture Models	7	Densities	One Mixture/Class, Weights
Perceptron	8	Weights	Weights/Layer, Firing Thresholds
Radial Basis Functions	8	Weights	Weights/Layers, Radii
Bayes Classifier	9	Densities	One-Dimensional Densities
Bayesian Network	9	Densities	Confusion Matrices

Table 27.1: Classifiers by Model Complexity.

ally, weights can be everything from hyperplane parameters to synaptic weights in neural networks. Ordered by complexity, thresholds are the simplest models, then come references and eventually densities. The rank of the weights depends on their concrete design.

As we can see from Table 27.1, some methods are co-ranked. Decision stumps and random selection are both very simple methods that require exactly one scalar as the model. The k-means classifier requires exactly the same number of references as the self-organizing map. Only the training process is more sophisticated in the latter method. Sometimes the complexity of the model depends on the topology of feature space. For example, a decision tree can be very simple or very complex – depending on the input data. However, in general, it is significantly more complex than random selection and less complex than k-means. This is equally true for the support vector machine and some other methods.

Micro Process	*Description*	*Concept Theory*
Best Fit	$\min(m(r, f))$	Prototype Theory
Comparison	$f < \epsilon$	Classical Theory
Maximum Likelihood	$arg\max_i P_i$	Theory Theory
Quantized Weights	$quant(w.f)$	Classical Theory
Similarity Measurement	$m(q, f) < \epsilon$	Prototype Theory

Table 27.2: Types of Micro Processes.

Table 27.2 lists the fundamental types of micro processes. So far, we encountered five methods. *Best fit* is typically applied in k-means and some related methods. We search for the best representative in a collection of references that represent the model. The best fit paradigm is based on the prototype concept theory. Out of a population of prototypes we select the best match (the *typical* one). If metric distances are used for selecting the best match, this approach is prone to being supersemantic.

Comparison is a straightforward approach in which we transform quantities into binary predicates that are interpreted logically. One typical example is the decision tree, in which a macro process of logical expressions is based on a comparison micro process. Comparison processes are usually not supersemantic and follow the (neo-)classical concept theory. Concepts are fenced off by thresholds.

The *maximum likelihood* principle needs no explanation. It covers all density-based methods and can equally be applied on a priori statistics and a posteriori knowledge. The Bayes classifier is a typical example. The norms (densities, mixtures) used for the representation of the description space indicate that the maximum likelihood principle is neither a straightforward implementation of the classical nor the prototype concept theory. It is rather an implementation of the theory theory in which norms stand for mental theories that are developed and refined over time.

Quantized weights is a two-step method. In the first step, the input is weighted and in the second the result is quantized. The example par excellence is the support vector machine. Input vectors are projected on the search space and then quantized to two classes by the separating hyperplane. The danger of supersemantic behavior in the maximum likelihood method and the quantized weights method is limited. As a typical micro process for separating classifiers, the quantized weights process is based on the classical concept theory.

Eventually, *similarity measurement* is the psychologically best understood micro process. Here, we measure the similarity between a stimulus and some reference (e.g. a query). Similarity measurement is, for example, employed in cluster analysis and the vector space model. Like for best fit, similarity

measurement can be perceived as supersemantic if inappropriate measures are used. This issue is discussed in detail in Chapter 28. Obviously, similarity measurement is a key ingredient of the prototype concept theory.

Classifier	*C*	*Training*	*Micro Process*
Random Selection	1	-	Comparison
Linear Discriminant Analysis	2	Density Estimation	Maximum Likelihood
Mixture Models	2	Density Estimation	Maximum Likelihood
Support Vector Machine	3	Optimization Process	Quantized Weights
Decision Stump	4	Heuristic	Comparison
Vector Space Model	4	-	Similarity
Bayes Classifier	5	Density Estimation	Maximum Likelihood
Bayesian Network	5	Density Estimation	Maximum Likelihood
Gaussian Bayes Classifier	5	Density Estimation	Maximum Likelihood
Decision Tree	6	Convergent Learning, Heuristic	Quantized Weights, Comparison
K-Means	7	-	Best Fit
Self-Organizing Map	7	Convergent Learning	Best Fit
K-Nearest Neighbor	8	-	Best Fit
Boosting	9	Convergent Learning	Quantized Weights
Cluster Analysis	9	-	Similarity
Radial Basis Functions	10	Convergent Learning	Quantized Weights
Perceptron	11	Backpropagation	Quantized Weights

Table 27.3: Classifiers by Micro Process (C)omplexity.

It is interesting that the wide variety of classifiers can be approached by this limited number of micro processes. Table 27.3 sets micro processes in relation to classifiers and describes the type of training employed. The categorization methods are ranked by the complexity of their concrete micro processes. Random selection is on the top, because it is based on simple comparison of the input data to some random reference. The standard neural network can be found at the bottom, because it requires (multi-layer) weighting of the input.

The table shows a trade-off between training complexity and application complexity. Complex models and training processes (e.g. density estimation) allow for quick application of the micro process. The maximum likelihood principle can be executed quickly. On the other hand, the simple or non-existent

training processes of the k-means classifier and cluster analysis make complex micro processes necessary. We conclude that – as so often in computer science – storage (complex models) can be traded for processing power (complex micro processes). The top-ranking of the density-based methods justifies our view that these methods are tailor-made for implementation in limited environments (embedded systems, mobile systems). See Chapter 10 for details on this issue.

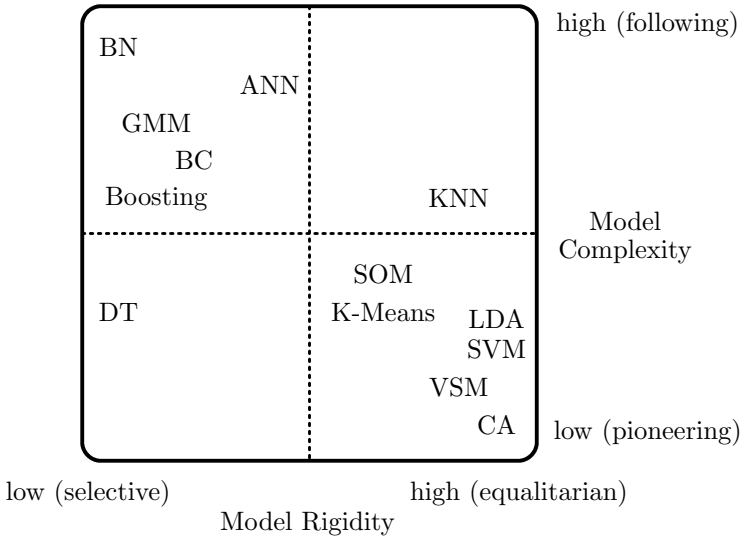


Figure 27.1: The Categorization Portfolio.

Figure 27.1 organizes the classifiers in a portfolio. We reuse *model complexity* from Table 27.1 as the vertical dimension. The rigidity of the model (flexibility) is drawn on the horizontal axis. Rigidity can be seen as the inverse of proneness to overfitting. As the overall layout shows, there is a correlation between model complexity and rigidity. Some models are too simple for being flexible and other models pay their flexibility with high complexity.

We call the classifiers with simple models *pioneering* and the others *following*. Classifiers with flexible models are called *equalitarian*, the others *selective*. Selective classifiers are distinguished by their dependence on appropriate input. For example, cluster analysis is a selective pioneer. This method can very well be used to get a first overview over feature space but it will produce results of little use if no clusters of medium size do exist in the data. In contrast, decision trees are equalitarian pioneers, because they are able to separate any data well. On the other hand, decision trees are prone to overfitting which makes them questionable for the eventual use in media understanding applications.

The k-nearest neighbor classifier is a selective follower. It can be (and is) used for categorization in media understanding applications but the quality of the results depends heavily on the ground truth, i.e. the labels of the references. A nice example for an equalitarian follower is the Bayes classifier. This method is applicable on almost any data set and fair enough for media understanding applications of a not too wide focus. In summary, we recommend pioneers for early application and followers for market ready applications. Selective methods perform well if the data fits the classifier. Then, they should be superior over equalitarian methods. However, on arbitrary data the latter type of method should perform superior.

The central machine learning question of media understanding is: When which classifier? The portfolio is intended to answer this question. With today's set of methods almost any ground truth can be represented. The relevant decision parameters are the *belief in the representativeness of the ground truth* which operationalizes as the *risk of overfitting* and the *effort required for training, model storage and application.* The experimenter needs to make up her mind about these issues in the first place. Then, the tables presented in this section can be used to select a few appropriate methods, test them and select the best-performing one.

27.2 Limits of Learning

Now, we move from concrete algorithms to the general frontiers of categorization. In detail, we investigate four limits of learning.

1. The *general feasibility principle* of learning
2. The *general convergence principle* of categorization machines
3. Quantization of the *general learning ability* of classifiers
4. Quantization of the *general training requirements* of classifiers

All of these limits are attributed as *general.* That is, we do not focus on particular algorithms but investigate the overall limits of the domains. It is true that some of the presented theories are closely linked to particular classifiers. For example, the convergence principle and the measure for learning ability were both developed by Vapnik who is also the father of the support vector machine. The theory of training requirements is related to the development of boosting. Still, the principles are also applicable to the other classifiers. Hence, we present them together as a handy theory of the learning process.

The feasibility principle of machine learning is simple: A learning algorithm is only relevant if it is able to learn all possible patterns expressible by a ground

truth (function) in less or equal polynomial time. Here, the ground truth function will only include patterns relevant in our world, not arbitrary ones. Still, reaching this degree of feasibility is hard or even impossible to achieve for some classifiers. Hence, the principle is often applied in relaxed form: *A learning algorithm is relevant for those learning problems which it can solve in less or equal polynomial time.*

The convergence problem was discussed in detail for information filtering procedures in the last chapter. Of course, it is of the same (or even greater) relevance for learning algorithms. Both the training process and the application process must converge. That is, the training process must finish in finite time and result in – at least – a local optimum. Oscillation (see next section) must be avoided. The application process must not show oscillating behavior as well. That is, the result of the application process must not depend on the time of termination.

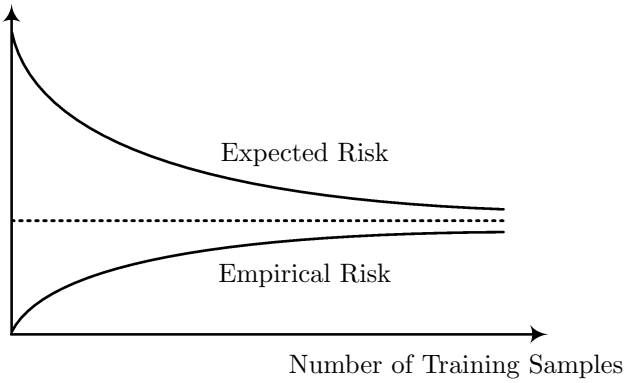


Figure 27.2: Consistency of the Learning Process.

The general model of convergence in classification was developed by Vapnik and is based on his notion of *expected risk* and *empirical risk* as laid down in Equations 18.1 and 18.5. The idea is illustrated in Figure 27.2. Over the training process, the quality of the classifier should approach the optimum indicated by the dotted line. The location is the tangent to both risk curves. The empirical risk rises with the number of training samples applied, but the rate of increase moves to zero. That means that we expect the learning process to improve categorization results over time. The classifier will fail less (often) later than earlier. On the other hand, the expected risk describes the application behavior of the classifier. In the early stage, errors will be larger than in later stages. We conclude that the expected risk can be seen as a function of the first derivate of the empirical risk. We will discuss this function at the end of this section.

The samples of the empirical risk approach the model of the expected risk. The optimum of both curves is the *best representation* that can be reached by the classifier under investigation.

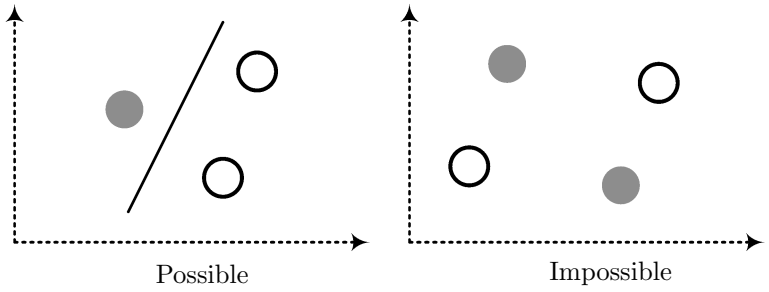


Figure 27.3: VC Dimension of the Separation Line.

How good can this optimum be? This question asks for the discrimination power of machine learning methods. A nice answer was proposed by Vapnik and Chervonenkis, hence it is called the *Vapnik Chervonenkis (VC) dimension of learning*. The VC dimension h of a classifier is the *number of data points that can be separated by it in arbitrary fashion*. Figure 27.3 shows an example. The straight line (e.g. the discrimination function of a perceptron) is able to separate three data points, but not four. The example on the right side is not separable by a straight line. A density model (e.g. a mixture) would be able to separate both examples.

The VC dimension is a straightforward practical approach for the description of discrimination power. However, there are two degrees of freedom. The first is the size of feature space. As we discussed in Chapter 18, a given number of data points can more easily be separated in a higher-dimensional space than in a low-dimensional one. There is simply more space that separates the data points. The second parameter of relevance is the number of classes that are allowed. The more possible groups the data points can form, the harder the discrimination problem becomes.

Practically, the VC dimension is often measured for two-dimensional feature spaces that contain a population that needs to be separated in two groups. For this problem definition, Table 27.4 lists the VC dimensions of the discussed classifiers. Cluster analysis has been added to the group with high discrimination power, because of the interpretability of the outcome. For the same reason, this classifier could be evaluated to $h = 1$ (the absolute minimum). For k-means and the self-organizing map, we assume a codebook of k reference vectors. Each reference vector behaves like a vector space model, which is capable of separating three data points. The k-nearest neighbor algorithm has a VC dimension $h = 2$

Classifier	*VC Dimension*	*Overfitting Risk*
Decision Stump	2	low
K-Nearest Neighbor	2	low
Linear Discriminant Analysis	2	low
Support Vector Machine	3	low
Vector Space Model	3	low
K-Means	3K	medium
Self-Organizing Map	3K	medium
Bayes Classifier	∞	high
Bayesian Network	∞	high
Cluster Analysis	∞	high
Decision Tree	∞	high
Mixture Model	∞	high

Table 27.4: VC Dimension of Important Classifiers.

if $k = 1$. For larger k, h rises accordingly.

As we see from the table, three groups of classifiers emerge. The discrimination power of the linear support vector machine is not generally better than the one of the vector space model. K-means and self-organizing map have the same VC dimension despite of the different learning algorithms. Bayesian networks, decision trees and mixture models are able to model any groupings. There is a link between VC dimension and the risk of overfitting. The VC dimension stands for discrimination power and flexibility of learning. Flexibility in the learning process goes hand in hand with high risk of overfitting. Methods with a high VC dimension can be described as *believing* in the ground truth data while $h \rightarrow \min$ indicates a conservative classifier that believes in its model (experience).

The VC dimension can be used to estimate the boundary of the application error. The following equation from [381] describes the relationship.

$$r_{app} < r_{emp} + \sqrt{\frac{h\Big(\log\left(\frac{2n}{h}\right) + 1\Big) - \log\left(\frac{\epsilon}{4}\right)}{n}} \tag{27.1}$$

Here, r_{emp} is the empirical risk (training error), n is the size of the training set and ϵ is the confidence interval. With probability $1-\epsilon$ this relation will hold. The application error r_{app} can be seen as the aggregated expected risk. Hence, the second term of the equation stands for the shape of the transformation between empirical and expected risk.

From the application error we bridge to the last problem that we want to discuss in this section: How many training samples do we need to keep the

application error below an acceptable threshold? One answer to this problem is given by the *probably approximately correct learning theory* (*PAC theory*). PAC theory calls concrete classifiers and their parameterizations *hypotheses*. The central statement of PAC theory is a relation between training sample size and risk. Mathematically, the relation can be stated as follows.

$$\prod_k P_k\big(\text{classify}(x_k, a) = y_k \wedge a \text{ is wrong}\big) \leq (1-\epsilon)^n \tag{27.2}$$

That is, the probability that a wrongly trained classifier defined by hypothesis a classifies $k \leq n$ samples x_k correctly as y_k is smaller or equal the n times inverse confidence interval ϵ. This rule that appears trivial can be used to estimate the number of required training samples n. The desired confidence ϵ given, the probabilities P_k can be computed and aggregated until the desired quality level is reached. Parameter k is the iterator over the probabilities. The solution will be probably approximately correct.

The derivation of this rule goes as follows. We start with the hypothesis that a wrongly trained classifier (i.e. the rate of correct classifications) cannot be better than $1-\epsilon$. Hence, for the k-th description-sample pair we have $P_k \leq 1-\epsilon$. The formula is simply the multiplicative aggregate.

Practically, the left part of the relation is often estimated (not computed) in the interval $\delta \in [0, 0.5]$ while $\epsilon \in [0.001, 0.05]$. Then, the PAC n can be computed as:

$$n = \frac{\log \delta}{\log(1-\epsilon)} \tag{27.3}$$

In conclusion, there is no big theory yet that would describe the limits of machine learning. The presented models are established fragments of a future theory. VC dimension and PAC theory are of immediate practical use. The feasibility principle and the convergence principle are valuable background knowledge for the understanding of the machine learning problem. In the next section, we introduce another piece of background knowledge that will become valuable in the last section: major aspects of dynamical systems theory.

27.3 Dynamical Systems Theory

In Chapter 26, we stressed the importance of convergence for information filtering processes in media understanding. Convergence is also essential in the categorization step. The belief in the correctness of the categorization results depends primarily on the absence of oscillation, i.e. the class label, once determined, must remain constant. However, most categorization methods are in training and/or application dynamical systems. It is, therefore, not surprising

that they show under certain circumstances dynamical – even chaotic – behavior. For the safe application of classifiers in media understanding it is crucial to understand, when and where categorization processes converge, where they oscillate and if their behavior can even become chaotic.

This section and the next are dedicated to this true frontier of media understanding. Surprisingly, the behavior over time of classifiers has hardly been investigated in media understanding research so far. Categorization results are often taken as correct without critical analysis of the basis and the process of their computation. During our short journey, we will see that both the input data and the algorithm employed for categorization can cause undesired oscillating behavior.

The organization of the two sections is as follows. In this section, we introduce the workbench required for the analysis. We review the major properties of ergodic systems and outline the most important aspects of measure theory, which is required to define orbits and attractors in dynamical systems theory. In the next section, we apply this knowledge on our set of classifiers. We will see that some methods are prone to oscillation while others guarantee convergence for any input configuration.

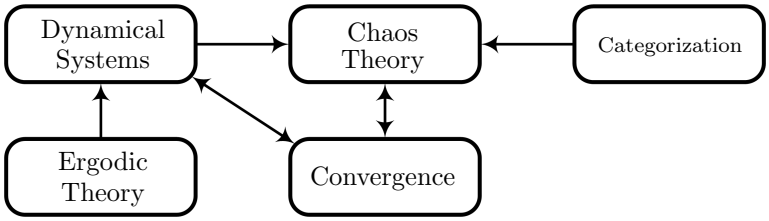


Figure 27.4: Categorization and Dynamical Systems Theory.

Figure 27.4 brings the topics of this section into context. The discussion focusses on the properties of dynamical systems. For that, we require basic knowledge of the structure of ergodic systems (open sets, σ algebras, etc.). From dynamical systems, stable orbits and attractors the trail leads on to chaotic systems with strange attractors. The tension between convergence, oscillation and chaotic behavior (e.g. bifurcation) is the theme of this section. The understanding of these terms is employed in the next section to investigate the training and application behavior of selected classifiers.

Dynamical systems theory deals with the behavior of *flow functions* over time. We want to know whether or not they stay in a limited orbit, an attractor. For that, we have to have an understanding of the *operations that can be performed by the flow functions* over time and we require a *method for measuring the membership* (distance) of a state of the flow to the attractor. The most

general description of the data manipulated in dynamical systems is arguably topological. Our basic unit of operation is the *open set*.

$$B_\epsilon(x) = \left\{ y \in X \middle| d(x,y) < \epsilon \right\} \tag{27.4}$$

The open set $B_\epsilon(x)$ is a sphere in the set X with center x and delimited by ϵ. A *closed set* would include ϵ in the sphere. Based on open sets, we can define a σ algebra Σ of space X by the following conditions.

$$A \in \Sigma \to X \setminus A \in \Sigma \tag{27.5}$$

$$\bigcup A_i \in \Sigma \tag{27.6}$$

$$\emptyset, X \in \Sigma \tag{27.7}$$

The requirements are unspectacular. The algebra must be closed over all subsets. The σ algebra can be used to define *Borel sets* as the smallest algebra that contains all open sets.

$$B = \inf \Sigma(X) \tag{27.8}$$

For our purpose it is sufficient to understand Borel sets as the basic building blocks of the space X on which the dynamical system under consideration (a classifier) operates.

That defines the data basis. Our second requirement is a measure that can operate on this data. General measure theory defines the following requirements for measures $m : \Sigma \to \mathbb{R}_{+,\infty}$.

$$m(A) \geq 0 \tag{27.9}$$

$$m(\emptyset) = 0 \tag{27.10}$$

$$m\Big(\bigcup A_i\Big) = \sum m(A_i) \iff \bigcap A_i = \emptyset \tag{27.11}$$

Any set in the algebra is projected on a positive real number. Empty sets are measured as zero. The union of non-overlapping sets is their sum. This straightforward definition of the set is, for example, fulfilled by the *Lebesgue measure* m_λ.

$$m_\lambda(B) = \prod (y_i - x_i) \tag{27.12}$$

Here, B is a Borel set. The pairs (x_i, y_i) define dimensions of the $i \leq n$ dimensional object represented by B. Hence, m_λ is a natural measure of the volume of regular objects. We will need it below for the definition of attractors.

With these requirements we can now define dynamical systems and, subsequently, ergodic (that is, working) dynamical systems. The central element is the *temporal flow function*.

$$\phi_t : X \to X \tag{27.13}$$

Here, X – the *phase space* – is a σ algebra of Borel sets on which measures m can be defined (e.g. the Lebesgue measure). Time is assumed to be discrete and positive ($t \in \mathbb{N}_+$). The flow ϕ_t has to have the following properties for sets $x \in X$.

$$\phi_0(x) = x \tag{27.14}$$
$$\phi_i(\phi_j(x)) = \phi_{i+j}(x) \tag{27.15}$$

Often but not necessarily, it will be C^k smooth and invertible as well.

$$(\phi_t(x))^{-1} = \phi_{-t}(x) = \bar{x} \tag{27.16}$$

Then, $\bar{x}$ is called the *preimage of the flow*.

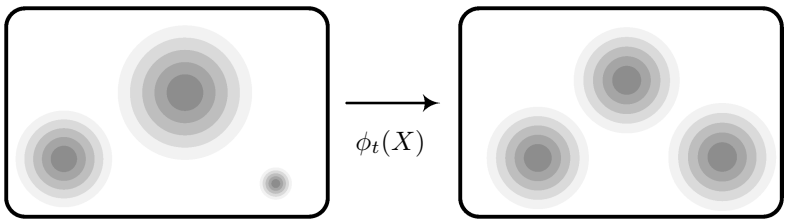


Figure 27.5: Example of an Ergodic System.

In our context, classifier training functions and application functions are flow functions that operate on description spaces and parameter spaces represented by σ algebras. Figure 27.5 illustrates a working system that operates on an algebra X. The flow function transforms one algebra of open sets into another. Formally, an *ergodic system* is defined as follows.

$$\phi_t(x) = x \to m(x) = 0 \vee m(X \setminus x) = 0 \tag{27.17}$$

Every set $x \in X$ that remains unaltered by the temporal flow must either be empty (e.g. of zero volume, if m is the Lebesgue measure) or comprise the entire algebra. Otherwise the system is not working (ergodic). Ergodic systems in this sense are, for example, the foundation of information theory (Section 20.3). Dynamical systems can be investigated for being ergodic by arbitrary measures except *invariant measures*. An invariant measure on a flow is defined as follows.

$$m(\phi_t(x)) = m(x) \tag{27.18}$$

That is, the measurement process is analogous to the flow operation. This can, for example, be the case for transformational similarity measures (see Chapter 28).

The *orbit* c of a dynamical system is defined as the set of points that can be reached by the flow function over time.

$$c(x) = \{\phi_t(x)\} \tag{27.19}$$
$$c(x) = \{x\} \tag{27.20}$$

The second equation describes a *rest position*. A flow function is called *T-periodic*, if $\phi_{t+T} = \phi_t$, i.e. the function oscillates in the orbit over the interval T.

Stability is defined for dynamical systems as for convergent systems and iterated function systems (last chapter): by *contraction*. We can use the Lebesgue measure to quantify the size of a set.

$$m_\lambda(\phi_t(x)) < m_\lambda(x) \tag{27.21}$$

Hence, the flow function contracts the set x. That will be the case if the Jacobi matrix of the flow function (first derivations in all directions) has a determinant smaller than unit size. A contracting flow function is called *dissipative*.

Eventually, a dissipative flow function in some orbit c will reach an *attractor* set $\bar{c}$. Attractors are characterized by the following properties.

$$\bar{c} \in X \tag{27.22}$$
$$\bar{c} = \phi_t(\bar{c}) \tag{27.23}$$
$$\bar{c} = \left\{x \in B_\epsilon(\bar{c}) \middle| \lim_{t\to\infty} \phi_t(x) = \bar{c}\right\} \tag{27.24}$$

The first requirement states that the attractor must lie in the algebra X on which the flow operates. The second one states invariance in the attractor against the manipulation caused by the flow function. Eventually, the attractor

must be closed. That is, in an open neighborhood of the attractor, the flow function must return from all[1] starting points to the attractor over time.

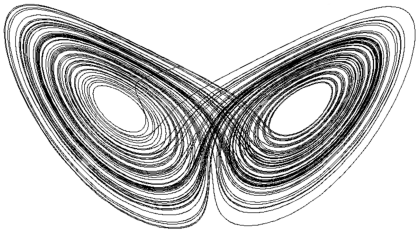


Figure 27.6: The Lorenz Attractor.

The general definition of the attractor covers different sorts of dynamical behavior. Points of rest are a trivial option. Oscillating flow functions can easily be covered by an attractor. But even *strange attractors* caused by *chaotic behavior* of the dynamical system are covered by the definition. Figure 27.6 shows the famous Lorenz attractor, the parameter space of a weather model, as an example. It is important to note that the existence of an attractor is no guarantee for reasonable behavior of the flow function (in our case, a classifier).

We require a notion of *reasonable* behavior. The straightforward solution is to call everything reasonable that is not chaotic. A chaotic dynamical system is according to Guggenheimer defined as *being sensitive to the initial state of the system.* More precisely, Devaney defines a chaotic flow function $\phi_t(x)$ with dense orbit c as follows.

$$m^{-1}\big(\phi_t(x), \phi_t(B_\epsilon(x))\big) \geq \delta \tag{27.25}$$

That is, the similarity between the set x and its open neighborhood will not converge. Measure m^{-1} is a distance function here. The definition covers the popular understanding of chaos as the behavior of a system in which a small change in the input can cause a large difference in the output. The resulting attractor is called a strange attractor.

Oscillating systems and chaotic systems require both feedback for the dynamical behavior. One form of chaotic behavior that is of particular interest is *bifurcation.* Bifurcation requires a dynamical system that can be parameterized. A bifurcation point lies at a parametrization where the characteristics of the flow function change dramatically. Sequences of bifurcations can lead to chaotic behavior.

[1] Precisely, *almost all*, i.e. all sets x with $m_\lambda(x) > 0$.

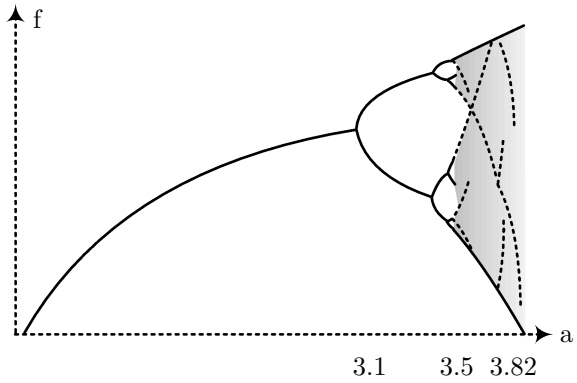


Figure 27.7: Bifurcation and the Logistic Map.

One famous example – next to the Mandelbrot set and Julia sets – of such a dynamical system is the *logistic map* illustrated in Figure 27.7. It is defined as follows.

$$x_{t+1} = a.x_t(1 - x_t) \tag{27.26}$$

Here, $x_t \in [0, 1]$ is the flow over the trivial algebra. The parameter a controls the behavior of the logistic map function. The function is typically used to simulate the development of a population. For $a < 1$ it will die, for $1 < a < 3$ it will end in a stable attractor, for $3 < a < 3.45$ it will oscillate between two values. The interesting behavior starts at $a = 3.57$. Then, the flow will end in a cascade of bifurcations (illustrated in the figure) and the development of the population will be chaotic.

This example should emphasize our statement from above: The existence of an attractor is no guarantee for convergent behavior. In particular, parameterized flow functions – such as classifiers – are often prone to oscillation or even chaotic behavior.

Before we conclude this section, we consider it beneficial to make a short note on the relevance of *game theory* for the judgment of the behavior of categorization methods. Generally, the situation in categorization is fundamentally different from a game. There is only one actor that tries to optimize his output. This actor does not have to consider the goals and strategies of competitors. One exception may be Bayesian networks where different theories compete for the explanation of an event or a sequence of events. Then, the investigation of the classification process by game-theoretic methods and the construction of Nash equilibria (optimal expected value for all players) may be of interest.

In this section, we have introduced a box of tools for the description and

analysis of dynamical systems. Flow functions, orbits, attractors, measures and ergodic systems are employed in the next section to analyze the dynamic behavior of classifiers in training and application.

27.4 Oscillating Classifiers

Below, we analyze the dynamic behavior of the self-organizing map, the expectation maximization algorithm in general and gaussian mixture models in particular. Furthermore, we investigate dynamics in associative memories, learning vector quantization and the Kalman filter. Primarily, we look for undesired oscillation in training and application that hinders a perfect convergence process. Alongside, we look out for chaotic behavior, namely bifurcation points.

Our method is simulation. For trivial settings (two or three states to simulate) the investigation has been done by hand. Larger parameter spaces and configurations were investigated by computer simulation. All parameter spaces and configurations (e.g. states) were traversed systematically. That is, for every dimension that required investigation, a linear sequence of samples starting from the origin and going to the maximum were considered. The curse of dimensionality limits this form of evaluation. For example, five dimensions with ten selected values each amount to 10^5 combinations that have to be investigated. Hence, we only performed simulations that end in acceptable time (a few hours to a few days). Despite the limitations, this form of simulation allows to provide a fair overview over the behavior of the investigated functions. Only few locations of interest will escape this statistical approach.

The terms introduced in the last section are used in the following way in this analysis. The learning algorithms and application processes are interpreted as flow functions that operate on a classifier model and parameter space (the σ algebra). By oscillation we mean a T-periodic flow function that returns to the starting point after T iterations. The flow function is considered to be ergodic, i.e. the classifier model is changed in each iteration of the learning/application process. However, convergence limits the magnitude of the ergodic process over time.

Which classifiers are dynamic? Only those that use an iterative process for training and/or application. Hence, methods such as the k-nearest neighbor classifier or the vector space model are not of interest. Furthermore, the optimization algorithms can be sorted out (support vector machine, decision tree, boosting). Eventually, the linear algorithms used in Bayesian networks provide no room for oscillation. Related methods such as the Bayes classifier can also be ignored.

What remains? Candidates for oscillation are the self-organizing map, the expectation maximization algorithm and mixture models. Joint with the con-

vergent filters, we investigate the following groups below.

1. Associative memories (Hopfield network, Boltzmann machine)
2. Self-organizing methods (self-organizing map, learning vector quantization)
3. Expectation maximization (including the Kalman filter)
4. Mixture model classifier

The convergence behavior of the associative memories has already been discussed in the last chapter. It is interesting that the same fundamental oscillation pattern exists both in the Hopfield network and the Boltzmann machine. It can already be shown in a network of just two nodes ($D = 2$). For two nodes s_1, s_2 and weights $w_{12} = w_{21} = 0.5$ and a firing threshold of $\epsilon = 0.5$, oscillation is the result of the following configuration.

Time	s_1	s_2
0	-1	1
1	1	-1
2	-1	1
3	...	...

Table 27.5: Oscillation in the Hopfield Network.

This T-periodic ($T = 2$) pattern can be extended to arbitrary dimensions $D > 2$ by selecting the weights between the old and the new dimensions small enough. This is just a simple oscillation pattern. We are positive that many other, more complex ones exist as well. However, we do not believe (and could not identify) chaotic behavior in the associative memories. The quantization space appears to be too limited.

Can the codebook of a self-organizing map – during training – become a strange attractor? Figure 27.8 illustrates idea. Oscillation of the codebook vectors would be caused by configurations of training vectors that draw the references back and forth to the same extent.

Our quantitative experiments suggest a negative answer to this question. Even in tailor-made setups, the adaptation of the reference vectors prevents oscillation. Depending on the center of gravity in the data, the movement of each reference keeps its direction from the first iteration on. Hence, it performs one-sided negative exponential convergence. We could not identify a configuration where the self-organizing map would have shown a different behavior. Hence, there is also no oscillation in the simpler learning vector quantization and since

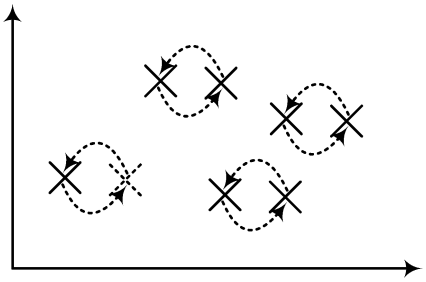


Figure 27.8: Example of an Oscillating Self-Organizing Map.

there is no oscillation there is also no room for chaotic behavior. Bifurcation points are anyway out of the question as there is no parametrization in the self-organizing methods. By the way, the self-organizing map shows the same convergence characteristic as learning vector quantization.

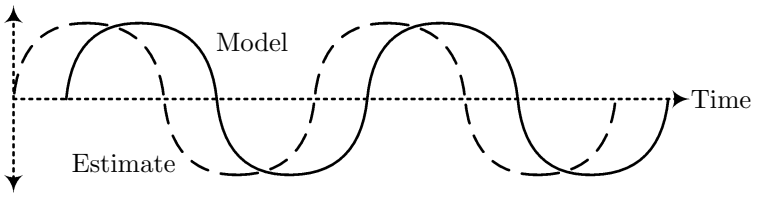


Figure 27.9: Oscillation in Expectation Maximization.

For the general expectation maximization algorithm, oscillation is confirmed. For example the *Lotka-Volterra equations* for the simulation of a population of foxes and rabbits oscillate in a stable fashion over time and have the same morphology as the expectation maximization algorithm. As we named it in the last chapter, it is a convergent process. This set of equations is defined as follows.

$$R_{t+1} = R_t(a_1 - a_{12}F_t) \tag{27.27}$$

$$F_{t+1} = F_t(a_{21}R_t - a_2) \tag{27.28}$$

Here, R_t, F_t is the size of the rabbit/fox population at time t, respectively. Parameter a_1 is the birth rate of rabbits, a_2 the death rate of foxes and the two other parameters describe how many rabbits become victims of foxes. The resulting oscillating process is illustrated in Figure 27.9. A large rabbit population provides the basis for growth in the fox population. The additional foxes will cause a reduction in the rabbit population, which in return makes life harder for the foxes, and so on.

What is in general true for the expectation maximization algorithm need not be true for the Kalman filter in particular. We could not identify oscillating behavior as clear as in the Lotka-Volterra model. Instead, we found a strong dependence of the Kalman filtering result on the initial state, i.e. the sequencing of the input values. Hence, oscillation in the Kalman filter is limited (among others, by the belief term/measurement update) but there is indication that the Kalman filter might show chaotic behavior if the input data is chosen appropriately. This issue requires further investigation.

Eventually, mixture models did not show oscillating behavior in our quantitative experiments, nor was there an indication of chaotic behavior. However, it appears from time to time that the mixture model learning algorithm does not terminate. The reason is that the threshold for termination is not met by the model. That is, it is not possible to describe the input data sufficiently exact by the model. This cannot happen for arbitrary mixtures but very well for relatively rigid models such as Gaussian mixtures. In case, it is recommended to relax the model and restart the training process.

Group of Methods	*Oscillation*	*Chaos*
Associative Memories	✓	×
Self-Organizing Methods	×	×
Expectation Maximization	✓	∼
Mixture Models	∼	×

Table 27.6: Results of the Dynamical Analysis of Classifiers.

Table 27.6 summarizes the results of our experiments (qualitative and quantitative) on oscillating classifiers. Associative memories and expectation maximization show oscillating behavior. Self-organizing methods – to the author's surprise – not. Chaotic behavior appears to be a minor issue in classification. However, the last word has not been spoken on this matter. Chaos, in particular, bifurcations are often hidden at particular locations of the parameter space. The employed simulation method gives a fair chance of identifying the majority of such points, but there is still the possibility of a miss.

We consider the dynamical investigation of practical classifiers an important frontier of future media understanding research. We have to perform deeper investigations of the named classifiers in order to understand better, under which conditions the training data and/or parameters will cause oscillation – or even chaos. Media understanding depends on the categorization method for the contextualization of the media summaries. This step is of paramount importance for the entire scheme. It must not be non-deterministic or even chaotic.

In the last two chapters we made the transition from media description to

media categorization. In the two remaining chapters we aim at setting media understanding in the human context. The next chapter deals with categorization processes, in particular, the micro process and its relationship to human similarity perception. Eventually, Chapter 29 deals with the apparent question: How can we do media understanding in the same way as humans?

Chapter 28

Human-Like Similarity Perception

Explains distance-based similarity, the improvements reached through the usage of predicate-based models, their integration in dual process models and the new perspectives gained from structural alignment and transformational similarity.

28.1 Similarity as Measurement

This chapter is about the micro process of categorization. Similarity measurement is the essential ingredient of two of five micro processes that were listed in Chapter 27. Performing similarity measurement is easy for humans but hard to imitate for machines. Below, we review the four major similarity theories that were mostly developed in psychological research. Furthermore, we endeavor to sketch a scheme for unification of these theories. This section focusses – besides the introduction of the general problem – on the classical similarity measurement path in which it is operationalized as distance. The next section rises from this semantically low level to the heights of predicate-based similarity measurement. We review the major psychological findings about the shortcomings of distance-based measurement and explain the psychologists' remedies. Section 28.3 deals with the merging of distance-based and predicate-based similarity measurement in dual process models as they were suggested in the last decade. Eventually, the fourth section introduces the two most recent similarity theories: structural alignment and transformational similarity. We embed these approaches in a general model in which the dual process model is the central component. In

summary, the goals of this chapter are *explaining the major similarity theories* and *unifying them in one model*.

Why is it important to have a similarity measurement model that is equivalent to human similarity perception? The major reason is that similarity measurement is such an important tool for human beings. The history of its scientific investigation goes back as far as Plato's image theory (Chapter 22). We have already sketched the various applications of similarity measurement in media understanding (which is a human-centered discipline). Other application domains in computer science include case-based reasoning, indexing and general signal processing. Outside computer science, similarity-based reasoning is employed in a number of research areas. In biology, for example, taxonomies are used to structure flora and fauna. Structural alignment is used to identify similarities in gene strings. In chemistry, the similarity of molecules is investigated, e.g. by maximal common substructures. In mathematics, we have similarity transformations, projections, the similarity principle of pseudo-analytic functions and differential equations. In medicine, homeopathy is based on the similarity principle. In physics, we measure the similarity of linear and exponential processes and of thermodynamics. In electrical engineering, we have the similarity principle of gas discharge and the similarity theorem of the Laplace transform. In mechanics, we have the affinity laws and the law of dynamic similarity. And so on. In everyday life, we orientate ourselves by object similarity, people with similar ideas are sympathetic, we want to be dressed similar to certain sports and movie stars, etc. In short, similarity assessment is ubiquitous in the human world.

Therefore, it should be clear what similarity is. But that is not the case. In his *Seven Strictures on Similarity*, Goodman points out that '*similarity alone might be taken to be an empty explanatory construct.*' The listed applications of similarity are indeed heterogeneous. The terms are based on different concept theories. The measurement is often based on different norms. The principles of comparison can range from exact matches over categorization to the diffuse understanding of analogies. And we do not know much about the cognitive processes that cause human similarity perception. Hence, it is no surprise that we have a handful of – partially, conflicting – similarity theories but no commonly agreed understanding of what human similarity measurement is and what human-like computational similarity measurement should be.

It is the declared goal of this chapter to contribute to pushing this frontier of media understanding research (and many other disciplines). Our main focus is on the categorization problem and there on the micro processes that employ similarity measures (*best fit* and *similarity measurement*). For our approach it is important to understand that, essentially, categorization is an iterative process of *choice* and *learning*. Similarity measurement is performed in the choice step. We already encountered choice models in Chapter 17. Luce's model

is based on similarity by distance measurement. More advanced models, such as Shepard's add a generalization function (discussed in the same chapter). Such models are structurally similar to kernel functions – which is no surprise since both types of functions have the same task to perform and employ the same strategy: reordering of feature space. Below, we return to these concepts (choice model, generalization, etc.) only where absolutely necessary. Due to the size and importance of the investigated problem and the limited space available we will strictly focus on the core topics of similarity measurement.

The remainder of this section is structured by four topics. First, we discuss the idea of distance-based similarity measurement and introduce the necessary prerequisites (in particular, scales of measurement). Then, we introduce and structure the most relevant distance-based measures and measure groups, followed by a brief discussion of advanced topics of generalization. Eventually, we explain psychological experiments performed on distance-based similarity measurement models and summarize the shortcomings that were identified.

Why is similarity measurement a frontier of media understanding research? Because it is of paramount importance for the acceptance of such systems. Man is the measure in media understanding. We cannot afford to employ machine learning techniques for categorization that are subsemantic or – more frequently – supersemantic. The results produced by such a system would not be appreciated by the users. Surely, similarity measurement should be a general topic in machine learning. But it is the application that generates the need for a good solution. Hence, the general machine learning frontier is in every case also a media understanding frontier.

The fundamental idea of *similarity judgment by distance measurement* is simple: *Two stimuli are the more similar the smaller the difference between them is.* This idea works very well for perceptual (for example, visual) stimuli but partially as well for abstract stimuli. However, the general notion of similarity is influenced by a number of factors, of which the following list names the most important ones.

- Task (categorization, matching, analogical reasoning, etc.)
- Nature of the stimuli (abstract, perceptual – psychophysics!)
- Beliefs and norms of the individual
- Ability for generalization (curve characteristics)
- Personal choice model (similarity-based, grouping, etc.)
- Form of individual thinking: taxonomic, thematic
- Employed perceptual space and distance measure

- Employed scale of measurement (interval, nominal, etc.)

Some of the points of the list have already been discussed. Task types and types of stimuli in the first part of the book, choice models and generalization in the second part, beliefs and norms in the second and third part. Taxonomic versus thematic thinking was briefly mentioned in Chapter 22. We will return to this issue in the third section of this chapter. Generally, we can say that similarity judgment is situation-dependent, relative, role-based and context-based.

Two technical prerequisites of distance-based similarity measurement are the employed *spaces and scales of measurement*. We need not review the mathematical theory of vector spaces here. It is sufficient to state that under a *perceptual space* we understand a vector space that allows to perform some form of measurement. The measurement can be *metric*, then we speak of a *flat* or *Euclidean* space. It may also be non-metric in some form or other. The various options are discussed in the last part of the section.

Name	*Definition*
Nominal	$A = \{a_i \mid i \in I\}$
Ordinal	$B = \left\{ a_i \in A \middle\vert \bigwedge_{i,j \in I} < (a_i, a_j) \in \{true, false\} \right\}$
Interval	$C = \left\{ a_i \in B \middle\vert \bigwedge_{i,j \in I} -(a_i, a_j) = \lvert a_i - a_j \rvert \right\}$
Ratio	$D = \left\{ a_i \in C \middle\vert \bigwedge_{i,j \in I} : (a_i, a_j) = \frac{a_i}{a_j} \right\}$

Table 28.1: Scales of Measurement.

The fundamental scales of measurement were already introduced in Chapter 7. There, we explained the principal differences between the scales. For the discussion in this chapter – in particular, in Section 28.3 – we require a deeper understanding of scales. That is why Table 28.1 lists set-based definitions of the four basic scales of measurement. Here, I with $|I| = n$ is the *index set* of the points of measurement on the scale.

The four scales are defined by operators. The nominal scale is just a set of measurement points. The ordinal scale is a nominal scale that provides at least a comparison operator. For the sake of convenience, we could also defined an equality operator. However, equality can also be reached by symmetric comparison of two index points. The comparison operator is the micro process of the ordinal scale. Repeated application allows to sort the input nominal scale A. Please note that membership of the operations result is not a condition of the scale definition!

Similarly, an interval scale is an ordinal scale in which a contrast operator is defined. Again, the difference between two index points needs not be a member of the scale. This is also true for the ratio scale where we require an additional operator for relativeness.

Name	*Example*	*Sets*
Nominal	*Members*, e.g. Car Brands	Predicate Scale: $\{a_1 = 0, a_2 = 1\}$
Ordinal	*Sortables*, e.g. Marks	Alphabet
Interval	*Countables*, e.g. Fingers	$\mathbb{N}, \mathbb{Z}$
Ratio	*Qualifiables*, e.g. Temperature	Absolute Scales, $\mathbb{Q}, \mathbb{R}$

Table 28.2: Examples for Scales.

Table 28.2 lists a few examples for the scales of measurement. Practical nominal scales are all lists of membership, for example car brands, a list of personal friends and the things in a fridge. Fundamental sets that measure on ordinal scale are the scale of predicates (equivalent to the binary code). The alphabet is – for sorting – often considered to be scaled on the ordinal level. At least, we refer to it as ABC but never as MBX. The set of natural numbers is a typical example for an interval-scaled set. Absolute scales are ratio scales with a natural origin, for example, the temperature scales and the scale of speed.

Why are the scales of measurement relevant for similarity measurement? Because the scale of measurement of the dimensions of feature space determines the measurement process. For example, distance-based similarity measurement operates always on data that is at least interval-scaled. Predicate-based measures operate exclusively on the predicate scale. It will be the topic of the third section how we can overcome the limitations introduced by the scales of measurement in the input data for better unified similarity measurement. Generally, it is helpful to perceive similarity measurement on interval/ratio-scaled data as *quantitative measurement* and on nominal/ordinal-scaled data as *qualitative measurement*, because in the first case the output is a quantity and in the second case a predicate (quality).

We have already discussed a number of distance-based similarity models in various chapters of this textbook. They are all listed in Appendix B.1 and can be organized into the following groups.

- True distances (mathematical and psychological Minkowski distances, Mahalanobis distance, etc.)
- Vector products (dot product, cosine measure, Tanimoto index, etc.)
- Supremal distances (dynamic association models, Mallows distance, etc.)

- Divergences (Hellinger, Kullback-Leibler, Bhattacharyya, etc.)
- Correlations (correlation coefficient, Cohen's measure)

The true distances are the arguably most important group. The Minkowski distances were already mentioned a couple of times. The mathematical form ($a_1 = a_2$ in Q1) has been applied in numerous media understanding applications. Psychological Minkowski distances ($a_1 \neq a_2$) have recently been suggested to overcome the supersemantic behavior of mathematical Minkowski distances (see the end of this section). The common drawback of most true distances is their foundation: The metric axioms have partially been falsified for human similarity measurement. Hence, these models are practically important but psychologically questionable.

The vector products represent the other end of the scale of distance-based measures. In fact, they do not measure distance but similarity directly. Hence, the psychologically correct application of vector products does not require the introduction of a generalization function. The most important representative is the dot product, which may be seen as the inverse of the first-order Minkowski distance. The cosine distance is a close relative. The Tanimoto index uses another interesting normalization.

Supremal distances embed the actual distance measurement process in an optimization meta-process that tries to identify the supremum of some association problem. Of course, all of the dynamic association models discussed in the first part of this book belong to this category. Furthermore, the template matching measures discussed in Chapter 24 belong mostly to this category. The characteristic feature of the supremal distance measure is the optimization criterion, not the actual measure – which may, in fact, be a similarity measure.

Divergences are distinguished by the weight that is laid on the compared vectors. A divergence will focus on one stimuli and evaluate the similarity of the second to the first. Hence, a divergence measure will generally not be symmetric (as required by the metric axioms). Since human similarity perception has also been proven not to be symmetric, divergences are interesting options for the semantic representation of human similarity judgment.

Eventually, correlation functions focus not on the individual value but on the distribution over the scale, and they measure the likeness of these distributions for two stimuli. The typical example is the correlation coefficient which has its optimum in the middle of two extremes.

The practically most relevant items of the list in Appendix B.1 are the measures Q1 to Q5, Q8, Q9 and Q12. It is recommended to consider these measures for quantitative similarity measurement applications. Generally, the symbols x, y stand for two stimuli (description vectors) where every description element x_i with $i < K$ is at least interval-scaled. In addition, of the supremal measures

listed in Appendix B.3, measure M7 is of paramount practical importance. In particular, in the form of the earth mover's distance it is employed in numerous media understanding applications.

Distance measures can be applied directly or with a generalization model. In the first case, true distances are transformed to similarities by taking the inverse. Vector products can be employed unaltered. However, psychological experiments have shown that weighting based on a (disputed) generalization curve is an important feature of human similarity cognition and learning. Hence, it is always advisable to employ a generalization function on the distance score in order to arrive at a semantic similarity score. The current model of choice is the one proposed by Tenenbaum (see Chapter 16).

Alternatively, some authors suggest density-based models for distance-based similarity measurement. We have already mentioned the Krumhansl model (M1) in Chapter 17. Another interesting approach was suggested by Ashby and Perrin [10] who do not consider concrete distance measures but just the *perceptual effect* of *stimuli confusion*. Confusion is a form of false choice caused by high density in the area of the reference stimulus. The authors consider the similarity arbitrary in the individual experiment. Similarity is defined as the aggregated inverse empirical risk (danger of a loss). For generalization-like weighting of the risk, the authors suggest the natural logarithm.

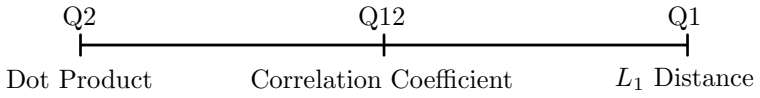


Figure 28.1: Scale of Quantitative Similarity Measures.

Before we continue with the critical evaluation of the appropriateness of distance-based similarity measurement, we would like to summarize the space of distance measures in a one-dimensional scale. Figure 28.1 provides a scale of quantitative similarity measures which is based on the following experiment. In spaces of m to n dimensional spaces we have computed the distances for the measures given in the appendix from each vector to all other vectors. The results were aggregated by statistical moments. The figure shows the average distance values for three representative measures.

As can be seen, the scale is spanned by the measures Q1 (city block distance, first order Minkowski distance) and Q2 (dot product). The dot product is a vector product, hence, a similarity measure. Since Q2 produces the extremal similarity score of all considered measures, we name this end of the scale *perfect similarity*. The L_1 distance defines the other end of the scale. No other measure produces a more extreme mean distance. Hence, we call this end *perfect distance*. In-between these ends we find all other distance measures from Appendix

B.1. For example, the L_2 norm (Euclidean distance, second order Minkowski distance) can be found immediately left of Q1. Interestingly, our systematic experiments located the correlation coefficient exactly in the middle of the scale. The variances are generally low ($\leq 10\%$).

Since none of the measures in the appendix exceed Q1 and Q2, it is tempting to re-define them as a linear combination of the extremes. Since Q2 is a similarity measure, the parameter on the scale provided in the figure defines the level of generalization for the distance measures. That is, generalization can be seen as an implicit function in the various groups of measures on the scale of quantitative similarity measurement.

The scale of quantitative similarity measures represents a nice, conflict-free world of similarity perception. However nice, this view is unfortunately not perfectly adequate. Psychological experiments have shown that human similarity measurement is only under specific conditions performed as described by distance measures. Most of the time, humans deviate significantly from this path. In the remainder of this chapter, we explain the experiments that have been conducted by psychologists, summarize their findings and how they correspond to the properties of perceptual spaces. This discussion provides the bridge to the next section, in which we discuss the remedies for overcoming the supersemantic behavior of quantitative measures.

Reference Stimulus 1 Stimulus 2

Figure 28.2: The Triad: Which Stimulus is More Similar to the Reference? (© *CNBC*)

The foundation of many psychological experiments on similarity perception is the *triad*. Figure 28.2 illustrates an example. The test person is confronted with a reference and alternatives. He has to decide whether he considers the first or the second alternative more similar to the reference. In the figure, we show a particularly tricky triad. The similarity of the reference to the first stimulus can be described as *thematic*: The scene shown in the video frame has the same layout, the same captions, etc. Only the anchor person is different. In comparison, the second stimulus is *taxonomically* similar to the reference. The situation is different, but some properties are the same (same anchor person, same topic, maybe same shot sequence, etc.). In Chapter 22 we introduced this

problem as the thematic taxonomic bridge. It is hard to decided which similarity is higher. We will return to this question in the third section of this chapter. There, we will see that the actual decision in this triad depends on the priming of the test person.

Such psychological experiments have been conducted since the 1930ies. Goldmeier was one pioneer. He found out that there is a major difference between the *attentive perception* and the *pre-attentive perception* of stimuli. In the first case, the shown signs are recognized and used for taxonomic comparison. In the figure, for example, the anchor person can be recognized by his face. Pre-attentive perception is performed by the human brain where no semantic stimuli exist or can be recognized. This is the case for *phenomenal* stimuli, as Goldmeier names them, but also for a rotated face. The human brain is not capable of recognizing all key features of rotated human faces.

Figure 28.3: Phenomenal Visual Features.

Figure 28.3 shows a few phenomenal features. Geometric primitives such as triangles and circles are typical examples. Right angles, parallels and symmetries also belong to this category. Goldmeier found out that the similarity of phenomenal features can best be described by distance measures. In the next section, we will set this in context with predicates, call such stimuli *integral features* and suggest the application of negative convolution (distance measurement). The view of Goldmeier has been confirmed by other authors, including Amos Tversky and Hubel and Wiesel, the pioneers of visual perception and cognition.

Goldmeier points out that the human brain reacts stronger on particular types of stimuli than others. For example, the measurement of vertical symmetries is more rigid than of horizontal symmetries. Attneave could show that the strongest integral stimuli are points of high curvature. We introduced this fact already in Chapter 14. It is the foundation of interest point detection in visual media description. Eventually, Tversky could show that Gestalt laws are another important ingredient of the similarity perception of integral stimuli. The *goodness of form* determines *diagnostic features*, i.e. stimulus properties that determine categorization.

However, psychological experiments have not just revealed where which distance measure should be used but also that in certain situations no quantitative measure is able to represent human similarity perception adequately. These experiments where usually performed in flat perceptual spaces in which the metric axioms hold. We introduced the metric axioms already in Chapter 8 as the

foundation of perceptual spaces. For example, the Minkowski distances fulfill the metric axioms. That is one reason why they are so popular measures in signal processing.

Name	*Definition*	*Criticism*
Identity of Self-Similarity	$m(x,x) = m(y,y)$	Larger Stimuli are More Self-Similar
Minimality	$m(x,y) \geq m(x,x)$	-
Symmetry	$m(x,y) = m(y,x)$	Complex Stimuli are Less Similar to Simpler Ones
Triangle Inequality	$m(x,z) \leq m(x,y) + m(y,z)$	Thematic Taxonomic Bridge

Table 28.3: Insufficiencies of the Metric Axioms.

Table 28.3 lists the metric axioms for stimuli x, y, z and a distance measure m as well as their essential insufficiencies with respect to human cognition. The only unchallenged axiom is minimality. Every pair of different stimuli will be more dissimilar than one stimulus to itself. Usually, we define the distance $m(x,x) = 0$.

The three other axioms have all been falsified for important types of stimuli. In particular, Tversky could show that they do not hold for abstract nor visual stimuli. The identity axiom does not hold if a larger stimulus x with more signs (hot stimulus) is compared to a smaller y with less signs (cold stimulus). Then, $m(x,x) < m(y,y)$ for most test persons. That is, if a stimulus shows more details, the level of similarity is increased. Hence, the distance is smaller than for a stimulus with few details. This behavior can be explained as moving away from arbitrariness or: the *inversion of entropy*, as Flusser called it.

The symmetry axiom does not hold for the comparison of complex stimuli x to simple stimuli y. Then, $m(x,y) > m(y,x)$, i.e. the similarity of the simpler stimuli to the more complex one is higher than the other way round (and inverse for the distances). Tversky named as an example the similarity of China (x) and North Korea (y). Most test persons found that North Korea is more similar to China than China to North Korea. This behavior can be explained by the human desire for more specific concepts. As we stated in Chapter 23, human beings are prone to believing in more specific joint events. The violation of the symmetry axiom expresses this behavior.

The most clearly falsified metric axiom is the triangle inequality. It is illustrated in Figure 28.4. The axiom appears natural. In a flat space it is not possible to reach p_2 from p_1 on a shorter path than the straight line. No path over some point p_3, p_4, p_5 that is not on the geodesic line will be shorter. However, human similarity perception is obviously not flat. For example, there is the problem of

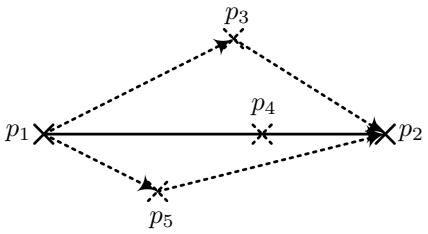


Figure 28.4: Example of the Triangle Inequality in an Euclidean Space.

the thematic taxonomic bridge. If we have three stimuli x, y, z that represent visualizations of a ball, a balloon and an airplane, then $m(x, y)+m(y, z) < m(x, z)$ for most test persons. Why is that? Because $m(x, y)$ is judged thematically (similar shape), $m(y, z)$ is judged taxonomically (similar property *can fly*). Both times the similarity is high, hence the distance is small. But the similarity between a ball and an airplane is very low thematically and taxonomically. We conclude that humans are able to measure similarity in different ways simultaneously. Metric distances cannot do that.

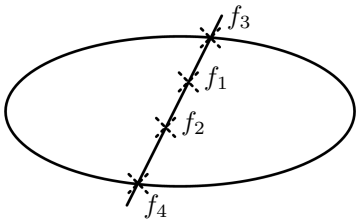


Figure 28.5: Cayley-Klein Geometries.

Several remedies have been suggested. The violation of the identity axiom can be ignored since it appears only for very specific cases. The symmetry axiom is ignored in quasi-metrics. The major problem is, therefore, the triangle inequality. In order to eliminate this insufficiency and still measure similarity as distance, it has been suggested to give up the *parallel postulate of Euclidean geometry*. Doing this forces us to enter the world of Riemannian geometry, where distance is defined as a tensor $m_{ij}(p)$ differently in every point p of the fundamental manifold. For spaces with constant curvature, Cayley and Klein have proposed a nice scheme for the definition of non-Euclidean *projective spaces* by the usage of conic sections and uniform distance measurement. Figure 28.5 shows an example. The fundamental ellipse defines the type of geometry (here, spherical). Distance is measured by the cross-ratio of co-linear points.

$$m(f_1, f_2) = a. \log \frac{(f_1 - f_2)(f_3 - f_4)}{(f_3 - f_2)(f_1 - f_4)} \tag{28.1}$$

Here, a is a global scaling factor (weight). The advantage of this model is that distance is measured without the limitations of the triangle inequality and that the type of geometry can easily be defined by the selection of an appropriate conic section.

Non-Euclidean distance models have hardly been investigated by psychologists so far and hardly been employed for distance measurement in practice. One reason might be that the Riemannian approach in its generality is very complex while the simple projective spaces do not allow for the definition of arbitrary distance measures. Hence, we are still searching for a better solution. The solution investigated by psychologists since the 1970ies is presented in the next section. For distance-based models we can conclude that they are good for integral stimuli, reliable, rational, but unlike human perception, supersemantic and, where not, inflexible. Despite these disadvantages, they are heavily employed in media understanding today.

28.2 Similarity as Counting

Reducing similarity judgment to a measurement process has proven inadequate for humans, as we saw in the last section. The alternative proposed by psychologists that would represent human similarity perception better is *counting*. In this section we review models and functions for the counting of the common properties of two stimuli (*communalities*) and of their differences. The organization is the same as in the last section. First, we introduce the idea, then we clear all necessary prerequisites, list and discuss the measures used in this area and, eventually, we group them and review the approach based on our findings.

The idea of similarity by counting is simple. We consider pairs of stimuli that are represented by distinguishable, nameable properties (signs). These signs are described by on-off values that indicate their presence in particular stimuli (e.g. 'man smiles,' 'car drives,' etc.). Such on-off values are called *predicates* or *taxa*. Similarity by counting is building a taxonomy on trees of predicates. The fundamental hypothesis of similarity by counting (*predicate-based similarity measurement*) is that similar stimuli will have many properties in common and only show few differences. The job of the predicate-based similarity function is to operationalize these *many*, *few*, etc.

It is interesting to note that the remedy for insufficient distance-based similarity measurement is based on a simpler scale: distance values are at least interval-scaled, predicates are nominal-scaled. However, this is only one side of the coin. The simpler items are loaded with higher semantics (e.g. *car*, *smiles*

above) and usually present in a larger number. As we explained in the first part of the book, predicates can be the result of iterations of media understanding. That is, the entire sophistication of a media understanding process is aggregated in the on-off values. Furthermore, predicate descriptions are usually longer than interval-scaled ones, i.e. feature space has a higher dimensionality. In summary, the structurally simpler predicates are by no means semantically simpler.

As motivation for the predicate-based approach to similarity measurement we would like to introduce the problem of *analogical reasoning*. Extensively discussed in psychology, there is no measurement-based solution to this problem of similarity perception. However, analogical reasoning can easily be modeled by common properties. All that is required is a semantic bridge from the concepts used in one part of the analogy to the concepts used in the other. Then, the alignment of the communalities is straightforward. In the last section we will see that this problem can also be solved by so-called transformational similarity models.

We have to discuss three foundations of predicate-based similarity measures. First, it appears advisable to investigate the major differences between distance-based measurement and predicate-based measurement. Then, we have to set predicate-based measurement in context with the problems of choice, norm theory and Bayesian inference. Eventually, we have to deal with the practical provision of predicates.

Taxonomic Thinking	*Thematic Thinking*
Steered by Surface Features	Steered by Deep Features
Considers Potential Similarity	Considers Psychological Similarity
Uses Qualitative Reasoning	Uses Quantitative Reasoning
Expressed by Predicates	Expressed by Quantities
Based on Separable Stimuli	Based on Integral Stimuli
Stimuli are Nominal-Scaled	Stimuli are Interval-Scaled
Stimuli have High Diagnosticity	Stimuli have Low Diagnosticity
Stimuli have Low Intensity	Stimuli have High Intensity
Represented by L_1 Norm	Represented by L_2 Norm

Table 28.4: Aspects of Taxonomic and Thematic Thinking.

Table 28.4 lists major differences of the predicate-based approach (left) and the distance-based approach (right) as they were identified by psychologists. In earlier chapters, we already introduced the terms *taxonomic thinking* for counting of communalities and *thematic thinking* for global similarity measurement. Hahn and Rascar have pointed out that taxonomic thinking is usually steered by surface features, i.e. features that are obvious for the semantic cognition of

humans. In contrast, thematic thinking is seen as the – often unconscious – processing of hidden features. Wallach named the first type of similarity *potential* (because it is immediately available) and the other *psychological* (because it cannot be defined/retrieved easily).

The fact that taxonomic thinking is based on counting is expressed in naming it *qualitative reasoning* while we called distance-based measurement *quantitative reasoning*. Below, we use these terms to distinguish the two approaches. Necessarily, taxonomic thinking requires *separable properties*, e.g. predicates. The quantities used for thematic reasoning are called *integral* features. The different scales required for these two types of properties were already discussed. Tversky points out that the *diagnosticity* (semantic value) will be higher for predicates than for quantities while it will be the other way round for their intensity (similarly to what McLuhan calls hot/cold). Eventually, Shepard has pointed out that the first-order Minkowski distance is a good measure for the imitation of taxonomic thinking while the Euclidean distance is adequate for quantitative reasoning. We will return to this idea below.

In Chapter 23, we introduced norm theory and the idea that norms represent human judgment about signs and relations between signs. Norm theory requires a rational foundation of signs for the construction of distributions that represent our model/experience of the world. Reasoning (e.g. similarity measurement) is inference from this model. The similarity of this concept to Bayesian inference is obvious. For example, mixture models employ the same model building, representation and reasoning principles. In fact, we can see Bayesian inference as an operationalization of norm theory. Predicates are in both models required for the representation of the basic set of symbols (signs, relations). Since they are available, they can of course be used for predicate-based measurement and inference. Hence, the measures introduced below can also be interpreted in a probabilistic context.

In Chapter 11, we introduced an iterative approach for the computation of predicates. In a first round of media understanding, proto-predicates are derived from integral descriptions by categorization. In further iterations, semantically richer predicates are derived. Rogers and Tanimoto proposed an even simpler approach. They suggest decision trees that split quantities into n on-off values. For example, the length of trees in meters can easily be transferred to three predicates *shorter than 1m*, *between 1m and 3m*, *longer than 3m*. In fact, this approach is a very simple implementation of iterative media understanding, in which a simple decision rule is used as the classifier.

The table in Appendix B.2 lists the remarkable number of predicate-based similarity measures that were suggested during the last 140 years. Please note that though the majority of the measures do, not all measures compute similarities. The measurement is always based on the communalities and differences of a pair of stimuli, which can be expressed by four variables a, b, c, d and the

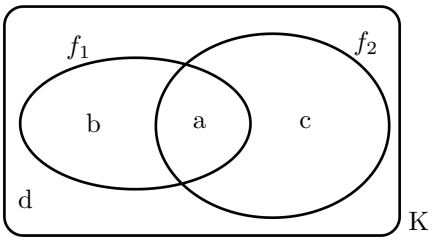


Figure 28.6: Types of Correspondences of Two Predicate-Based Stimuli.

dimensionality of the description vectors (number of predicates) K. Figure 28.6 explains the four types of correspondences for two media objects o_1, o_2 with descriptions f_1, f_2.

- Variable a counts the number of predicates that are present (1) in both objects (communalities).
- b counts the number of predicates that are present in o_1 but not in o_2.
- c counts the number of predicates that are present in o_2 but not in o_1. The sum $b + c$ stands for the differences between the stimuli.
- Variable d counts the number of predicates that are not present in both stimuli. Naturally, d is almost unlimited. Hence, this correspondence is only used in few measures.

Furthermore, $a + b + c + d = K$. Please refer to the appendix for the formal definition of the four variables. The foundation on just four variables allows for easier in-depth analysis of predicate-based measures than of quantitative measures.

Before we discuss the concrete measures, we would like to mention the attempt of Tversky to define an equivalent for the metric axioms in the world of predicate-based similarity measurement. The *monotone proximity structures* are a set of three axioms that have to hold for a predicate-based similarity measure in order to be reasonable. For objects o_i with two-dimensional descriptions (f_{i1}, f_{i2}) the first axiom – the *dominance relation* – is defined as follows.

$$m(f_{11}, f_{22}) < m(f_{11}, f_{12}), m(f_{12}, f_{22}) \tag{28.2}$$

Each similarity value that is measured over both dimensions (objects, predicates) must exceed the one-dimensional projections of the media objects and the description elements. This axiom measures a property similar to the triangle inequality, though it is less strict.

$$m(f_{11}, f_{21}) < m(f_{31}, f_{41}) \iff m(f_{12}, f_{22}) < m(f_{32}, f_{42}) \tag{28.3}$$

The second axiom is the *consistency axiom* shown above. It requires that the similarity measure must be consistent over all dimensions of feature space. Inversion is not allowed. This axiom appears surprisingly strict for a similarity measure. The practical applicability is questionable. The last axiom measures *transitivity*.

$$f_{11}|f_{21}|f_{31} \wedge f_{21}|f_{31}|f_{41} \Rightarrow f_{11}|f_{21}|f_{41} \wedge f_{11}|f_{31}|f_{41} \tag{28.4}$$

Here, $f_{11}|f_{21}|f_{31} \equiv m(f_{11}, f_{31}) < m(f_{11}, f_{21}), m(f_{21}, f_{31})$. This axiom integrates the AND-operator in the similarity measurement process. Logically connected border elements cover all inner elements.

The practical application of the monotone proximity structures is limited. Some of the measures in the Appendix (in particular, P6) fulfill the axioms, others not. The entire set of measures falls in three groups.

- *Co-occurrence measures* that emphasize the communalities of two stimuli.
- *Distance measures* that focus on the differences between the stimuli.
- *Contrasts* that try to establish a balance between communalities and differences.

Typical co-occurrence measures are P1, P2 and P5. These measures are strict similarity measures. The arguably best-known distance measure is the Hamming distance (P3). This measure is, for example, used in text understanding for the measurement of the dissimilarity of pairs of words (e.g. of a wrongly spelled word to its orthographically correct form). Another important distance measure is P8, which is often used in cluster analysis algorithms. Eventually, frequently employed contrasts are Tversky's feature contrast model (P6), P9, P11 and P21.

The feature contrast model P6 is of particular interest for us. It has been proposed by Tversky as a measure that fulfills the axioms of monotone proximity structures and that combines the advantages of taxonomic and thematic measurement. Tversky considered this measure the solution for the problems of distance-based measurement. However, as we will see below, predicate-based measurement – even the feature contrast model – cannot represent all aspects of human similarity cognition.

Closer investigation of the table of measures shows that the characteristics of a measure (in particular, a contrast measure) depend on the processing of b, c and the way communalities and differences are related to each other. Measures that use $b.c$ instead of $b+c$ will be sensitive for well-balanced differences between the stimuli while the others will produce linear difference sums. Similarly, the

contrast P7 reacts much stronger than P6 on imbalances between communalities and differences. The general rule is that multiplication/division will create an optimization criterion of squared shape (like in linear regression), while addition/subtraction will produce linear optimization criteria. The first form is more discriminative while the second is less prone to outliers.

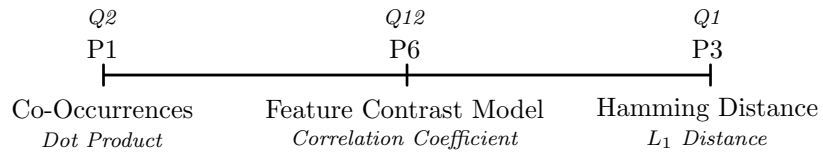


Figure 28.7: Scale of Predicative Similarity Measures.

Figure 28.7 shows a scale of predicate-based measures and equivalences with quantitative measures. This scale was computed in the same way as the one for distance measures in the last section. Systematic search through feature space (here, a, b, c, d) showed that the measures P1 and P3 are the endpoints of the scale and that all other measures can be positioned in relation to these two. In fact, each measure from the table in the appendix can be represented as a linear combination of the two extremes. For example, the feature contrast model P6 lies exactly in the middle of the scale. Most other contrast measures are located close to P6. The variances are generally low.

x	y	$x \cap y$	$x \setminus y$	$x.y$	x-y
1	1	1	0	1	0
1	0	0	1	0	1
0	1	0	0	0	0 (-1)
0	0	0	0	0	0

Table 28.5: Equivalences in Predicate Space between Operators for Nominal-Scaled Data and Interval-Scaled Data.

However, Figure 28.7 provides more. We propose equivalence between the co-occurrence counter P1 and the dot product Q2 (similarity measures) and of the Hamming distance P3 and the city block distance Q1 (for different scales, of course). This equivalence is given for predicate space – as shown in Table 28.5. For binary description elements (the table shows all possible combinations) the set operators and the linear/geometric operators are equivalent. In particular, $b+c = \sum |x-y|$. Hence, what we have is a scale of similarity measures that spans from taxonomic thinking (P1, Q2) to thematic thinking (P3, P1) and explains all other measures on the way. It can be applied directly on predicate descriptions

but as well on quantitative descriptions. Then, the description elements need to be normalized to the interval $[0, 1]$ and the quantitative measures (Q1, Q2) have to be used instead of the predicate measures. As we will see in the next section, this scale defines an interesting dual process model.

Before we move to the next section, we would like to point out that predicate-based similarity measurement helps to overcome the limitations of distance-based measurement but that the approach suffers from innate insufficiencies. In particular, cognitive scientists could show that humans employ not just taxonomic thinking even if they could. We use a mixture of taxonomic and thematic reasoning to derive our similarity judgments. Furthermore, psychologists could show that under time pressure taxonomic thinking is suppressed. Then, quick thematic judgments are derived from stimuli with high intensity. Eventually, some stimuli are integral by nature. Breaking such stimuli down into a sequence of predicates is unlike the processing performed by the human brain.

We conclude that predicate-based similarity measurement stands on a sound basis. The model is clear, well motivated and applicable wherever sufficiently semantic descriptions are available. For predicate spaces we could even show the equivalence of the space of qualitative measures and the space of quantitative measures. However, psychological research results show that qualitative measurement alone is not sufficient to measure similarity like humans do. What is required are models that combine the advantages of measurement and counting. Such models are discussed in the next section.

28.3 Dual Process Models

Dual Process Model (DPM) is a term used in psychology for a number of different phenomena. We focus on the meaning of *similarity measurement processes that employ taxonomic thinking and thematic thinking simultaneously.* In this context, a DPM is a meta-measure that uses a qualitative measure and a quantitative one. Below, we describe the major approaches that were proposed so far. First, we explain the idea and give some motivation. Then, we list and discuss the existing models. Eventually, we derive a DPM that appears – from the media understanding perspective – ideal and we explain its application.

It was an essential result of psychological similarity research of the late 1990ies that neither distance-based measurement nor predicate-based measurement can simulate human similarity perception satisfactorily. Wisniewski was one of the first to suggest the combination of thematic (for semantic properties) and taxonomic thinking (for syntactic properties, see Chapter 22) in the similarity measurement process. Navarro pointed out that proper similarity measurement requires both the consideration of discrete (separable) and continuous (integral) aspects. Simmons and Estes discovered that the combination of the-

matic and taxonomic thinking is not just an important part of human similarity judgments but that the relative share of each aspect is a characteristic property of the individual. That is, some people prefer rather taxonomic judgments while others prefer thematic judgments. Extreme forms (e.g. purely taxonomic thinking) hardly exist. There is a weak correlation between the form of similarity measurement and gender: Woman prefer thematic judgments, men prefer taxonomic judgments.

Aspect	*Taxonomic*	*Combined*	*Thematic*
Stimuli	Separable	Both	Integral
Descriptions	Local	Semi-Local	Global
Measurement	Dot Product	Correlation	L_1 Norm
Counting	Co-Occurrences	Feature Contrast	Hamming Distance
Convolution	Positive	*Dual Process Model*	Negative

Table 28.6: Context of Dual Process Models.

Table 28.6 sets dual process models in context with what we already know about human-like similarity measurement. Taxonomic thinking requires separable stimuli which are typically represented by local descriptions. The *scale of similarity measurement* suggests the dot product and the number of co-occurrences as taxonomic measures. These measures correspond with positive convolution. Negative convolution employs Minkowski distances or predicate-based measures like the Hamming distance. These measures are employed on integral stimuli represented by global descriptions.

Now, dual process models operate on both types of stimuli. The descriptions required for this process can be described as *semi-local*. A typical quantitative measure is the correlation coefficient. In the qualitative domain, the feature contrast model is representative. However, as we will see in the next paragraphs, other, more flexible dual process models have been proposed by psychologists and computer scientists that allow for the implementation of the individual taxonomic/thematic preference as a parameter.

The basic idea of the DPM should be clear by now: Combination of qualitative and quantitative measurement in one process, enriched by a parameter that expresses the individual preference for thematic or taxonomic judgments. Hence, dual process models require interaction with the user. The personal preference has to be identified by experiments (see below).

Before we go through the list of measures, we would like to point out that the definition of dual process models requires a solution for the scaling problem: Qualitative measurement happens on nominal-scaled data, quantitative measurement on interval-scaled data. In the last section, we showed that the application of quantitative measures on predicates is straightforward. But what

the other way around? The general solution is the application of *fuzzy set operators*. Such functions are ideal for the representation of quantities (probabilities, belief scores, etc.) in logical systems such as predicate-based measurement processes. Often, the required operators are defined for two vectors f_1, f_2 as follows.

$$f_1 \cap f_2 = min(f_1, f_2) \tag{28.5}$$
$$f_1 \cup f_2 = max(f_1, f_2) \tag{28.6}$$
$$\neg f_1 = 1 - f_1 \tag{28.7}$$

A concrete DPM that employs fuzzy set operators is the *fuzzy feature contrast model* (FFCM) discussed below. Before we come to this model, we would like to introduce the DPM suggested by Navarro. After the FFCM and its derivates we introduce the *quantization model* developed by the author.

Navarro's DPM is a simple psychological model that does not consider generalization. The similarity function is defined as follows.

$$m_{navarro} = m_{pred} - m_{dist} \tag{28.8}$$

Here, m_{pred} is an appropriate measure for predicate-based similarity measurement and m_{dist} is a measure for distance-based measurement. The measures have to be selected by the user and have to be appropriate for the types of stimuli that are processed. This DPM is a straightforward implementation of the idea of Wisniewski. It employs the distance term directly on the predicate-based measurement. Astonishing for a psychologically motivated model is that the model neglects generalization. This shortcoming limits the applicability of the model. It is mainly a showcase of the DPM idea.

The operators used in the fuzzy feature contrast model and the quantization model – two models defined by computer scientists – are described in Appendix B.4. The FFCM was defined by Santini and Jain as an extension of the feature contrast model (FCM) for the quantitative domain. Since the FCM is a basic DPM, the FFCM can also be considered being a DPM – though it was defined before the idea of dual similarity measurement entered computer science. The FFCM – in the corrected version of Tang, Fang, Du and Shi – employs the following operators for the representation of the components a, b, c of the FCM.

$$a = w_1 \sum \min(f_{1i}, f_{2i}) \tag{28.9}$$
$$b = w_2 \sum \max(f_{1i} - f_{2i}, 0) \tag{28.10}$$
$$c = w_3 \sum \max(f_{2i} - f_{1i}, 0) \tag{28.11}$$

Here, f_{1i} is the i-th element of the description of media object o_1. The w_i are weights that define the thematic taxonomic profile of the user. The actual measure is reached by applying these operators on the FCM model.

$$m_{\text{ffcm}} = w_1 \sum \min(f_{1i}, f_{2i}) - w_2 \sum \max(f_{1i} - f_{2i}, 0) - w_3 \sum \max(f_{2i} - f_{1i}, 0) \tag{28.12}$$

This version of the FFCM allows for asymmetric similarity measurement ($w_2 \neq w_3$) and arbitrary thematic taxonomic configurations. However, like Navarro's model, it does not consider generalization or the usage of arbitrary functions for quantitative and qualitative measurement.

The quantization model (QM) defined in [84] goes one step further than the FFCM by integrating the thematic taxonomic configuration in the definition of the operators for a, b, c, d. Instead of fuzzy set operators, the QM suggests statistical operators.

$$a = \sum s_i,\ s_i = \begin{cases} \frac{f_{1i}+f_{2i}}{2} & \text{if } \frac{f_{1i}+f_{2i}}{2} > 1 - \epsilon_1 \\ 0 & \text{else} \end{cases} \tag{28.13}$$

$$b = \sum s_i,\ s_i = \begin{cases} f_{1i} - f_{2i} & \text{if } 0 < f_{1i} - f_{2i} < \epsilon_2 \\ 0 & \text{else} \end{cases} \tag{28.14}$$

$$c = \sum s_i,\ s_i = \begin{cases} f_{2i} - f_{1i} & \text{if } 0 < f_{2i} - f_{1i} < \epsilon_2 \\ 0 & \text{else} \end{cases} \tag{28.15}$$

$$d = \sum s_i,\ s_i = \begin{cases} \frac{f_{1i}+f_{2i}}{2} & \text{if } \frac{f_{1i}+f_{2i}}{2} < \epsilon_1 \\ 0 & \text{else} \end{cases} \tag{28.16}$$

Here, the input descriptions are defined as in the FFCM. The two thresholds ϵ_1, ϵ_2 define the degree of thematic and taxonomic thinking. A high ϵ_1 emphasizes taxonomic aspects. If ϵ_2 is high, even large differences are considered, i.e. thematic thinking is emphasized.

The actual similarity score depends on the similarities and differences of the quantities and on the selected predicate-based measure. The QM can be combined with the FCM but as well with any other predicate-based similarity measure. In fact, it provides a kernel mapping for quantities that makes the actual similarity measurement a meta-process.

None of the before-mentioned DPM considers generalization. Neither FFCM nor QM allow for the usage of qualitative and quantitative measurement in one similarity function. Both methods provide just mappings from the interval scale

to the nominal scale. Since this state of affairs is unsatisfactory, we would like to suggest a DPM that covers the most important aspects of human similarity measurement.

$$m_{dpm} = a.m_{sep} + (1 - a)g(m_{int}) \tag{28.17}$$

Here, g is the generalization function (Gaussian, Tenenbaum or Shepard). The thematic taxonomic configuration is defined by parameter $a = \frac{k_1}{K} \in [0, 1]$, $k_1 + k_2 = K$ are the numbers of separable and integral description elements, K is used for normalization. The measures for separable and integral stimuli are defined as follows.

$$m_{sep} = \{P1, Q2\} \tag{28.18}$$

$$m_{int} = \{P3, Q1\} \tag{28.19}$$

That is, for qualitative measurement we allow measures P1 and P3 (without normalization). For quantitative measurement, Q1 (city block metric), Q2 can be used. Please note that parameter a defines the thematic taxonomic configuration of the user while the selection of P1/Q2 and P3/Q1 is just a technical issue determined by the scale on which the description elements measure. Linear combinations of the measures by a are able to represent any other form of similarity measurement. The actual selection of the measures will depend on the data types of the description elements. On predicates, the qualitative measures will be used, on quantities the others.

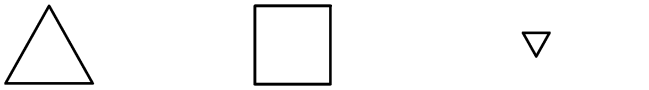


Figure 28.8: Thematic Taxonomic Test Triad.

The determination of a requires interaction with the user. We suggest the implementation of a short questionnaire of, for example, ten triads of the form illustrated in Figures 28.2 and 28.8. These triads force the user to decide whether she prefers thematic judgments (similar size in Figure 28.8) or taxonomic judgments (similar shape). After the test, the value for parameter a can be derived from the average response.

We are positive that the application of the proposed DPM will improve the quality of similarity measurement in media understanding applications significantly. The thematic taxonomic imprint of the individual user is the essential hidden parameter in this model. It has to be uncovered before it can be applied successfully.

28.4 Similarity as Alignment and Transformation

Dual process models are the similarity measurement procedures of choice today. However, alternatives/extensions do exit. In this section, we would like to discuss two models that are heavily discussed in psychological similarity research today: *structural alignment* and *transformational similarity*. In the first part of the section, we focus on alignment, in the second on transformation. As before, we start with motivation/idea, introduce representative models and end with a critical evaluation of the approach. This section closes with an *integrated model of the human-like similarity measurement micro process*.

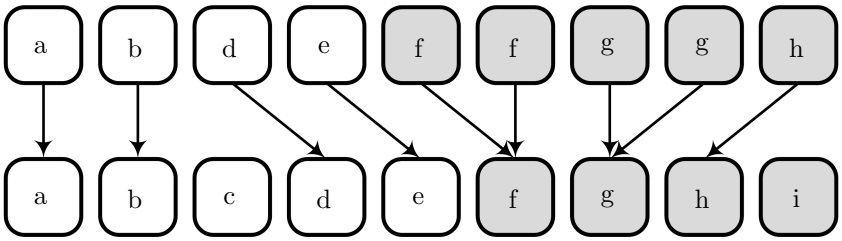


Figure 28.9: Structural Alignment Example.

Structural alignment was already discussed in Chapter 8. There, we named all relevant models and methods. The key idea is *homologization* as illustrated in Figure 28.9. Two *sequences of separable stimuli* are matched by *dynamic association* as good as possible. The actual process has to implement forms of insertion and deletion in a sequential optimization process – as, for example, in dynamic time warping.

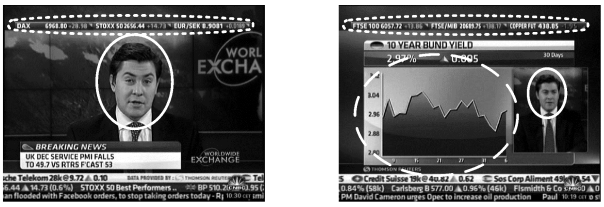


Figure 28.10: Communalities and (Non-)Alignable Differences (© *CNBC*).

The principal relationships of description elements in structural alignment are shown in Figure 28.10. *Communalities* are identical stimuli (signs) in both objects. For example, the face of the anchor person is a – taxonomic – communality of both of the shown video frames. *Alignable differences* are elements that are similar but not identical. The ticker on the top of the frame is an example

for an alignable difference: Though the content is different in the two frames, the concept is – thematically – the same. Eventually, *non-alignable differences* are elements that are present in only one object. In the figure, the chart is non-alignable as it is not present in the first object. Gentner and Markman have discovered that for human beings the major difficulty lies in the handling of non-alignable signs (neither taxonomic nor thematic). Furthermore, it has been shown that human beings employ a top-down strategy in structural alignment. Such a divide and conquer strategy is also recommendable form machine alignment (for example, as iterative media understanding).

The principal groups of structural alignment measures are probabilistic inference methods (see Chapter 9), similarity meta models such as the Mallows distances (e.g. Earth Mover's Distance), the Hausdorff distance and related supremal methods and graph matching (Chapter 24), which is likewise a typical application of structural alignment. Other important applications are the alignment of gene strings (e.g. Needleman-Wunsch algorithm) and of speech patterns (e.g. dynamic time warping).

The major shortcoming of structural alignment is – as Simmons pointed out – its impotence to explain thematic relationships in the input data well. Structural alignment requires separable stimuli on the level of the comparison process. that is, the method compares signs. It is a dynamic association process that employs similarity/distance measurement for the alignment of different signs but it is not able to analyze complex integral stimuli. Hence, structural alignment methods are not similarity measures but rather *similarity meta models* in which quantitative, qualitative and dual process models can be embedded.

The idea of *transformational similarity* is that the *degree of change required to transform one stimulus into another is a measure for their similarity/distance.* Transformational similarity has, for example, been tested for word similarities and artificial gene strings. The employed methods (e.g. edit distances) are structurally similar to the dynamic association models employed in structural alignment (e.g. dynamic time warping). The major difference is that transformational similarity is able to analyze (decompose) integral stimuli. In an iterative process, the first representation is transformed by *pre-defined operations* into the second one. That requires a sequence of analysis, segmentation, transformation and recombination similar to the ones employed in iterated function systems (see Chapter 25). In consequence, transformational similarity is well able to measure thematic relationships but hardly able to recognize taxonomic relationships.

There are two fundamental models of transformational similarity that should be discussed here.

- Levenshtein Metric
- Kolmogorov Complexity

The Levenshtein metric measures the distance from one stimuli to another by the number of *insertation*, *deletion* and *substitution* operations required for transformation. One implementation in text understanding is the *edit distance* that performs this process for strings (e.g. words). The application shows the general similarity of transformational models to distance measurement, as the alternative to the edit distance is the Hamming distance – a typical distance measure.

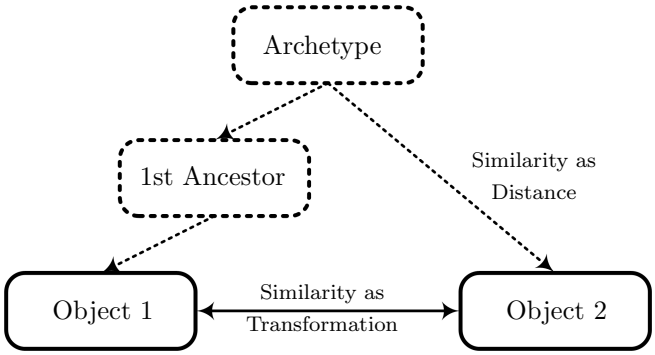


Figure 28.11: From Transformation to Distance Measurement.

The Kolmogorov complexity is the shortest binary program required for the transformation of one description into another. Alternatively, it can be defined as the shortest description of the data represented in two stimuli. Figure 28.11 illustrates both ideas. The ancestors of the two objects are created by transformation. The archetype is their common basis. While for the archetype view the solution is clear (the Lempel-Ziv-Welch algorithm for lossless compression), the solution for the best binary program depends heavily on the set of instructions. However, this view is a natural implementation of the idea of transformational similarity: Transformation is necessarily an active process.

The major problem of the transformational models is the absence of an undisputed set of operations. For example, it is not clear why the Levenshtein metric allows a substitution operator, which could also be implemented as a sequence of insertions and deletions. In the Kolmogorov complexity, the basis for data representation remains unclear. These degrees of freedom open a large space of possibilities – an example for the curse of dimensionality. Any selection of a point in this space must appear arbitrary. We doubt that a natural optimum exists. Even Hahn, who supports the idea of transformational similarity for text understanding, admits that the application of transformation models requires the existence of an accepted transformation vocabulary. Otherwise, the transformation will remain questionable.

A simple yet practical implementation of transformational similarity is the usage of iterated function systems for semantic comparison as described in Chapter 25. There, the contracting transform will be composed of the transformation vocabulary. The archetype will correspond to the self-similarity pattern T_0. Eventually, the number of iterations required for transformation is a distance measure for the compared objects (inverse self-similarity).

Before we conclude this chapter, we would like to remark that structural alignment and transformational similarity are very well suited for the evaluation of spatial relationships in, for example, visual data. Unfortunately, hardly any research has been performed in this area so far. However, we believe that the transformational approach with iterated function systems could be used to identify the essential signs that are then related spatially by structural alignment.

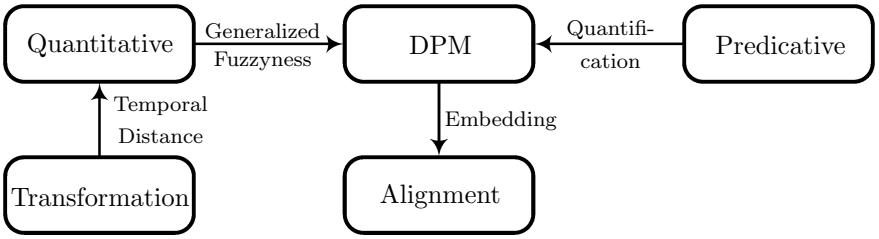


Figure 28.12: A Unified Similarity Measurement Model.

The conclusion of this chapter is a unified view of the similarity measurement micro process. Figure 28.12 illustrates how the parts fit together. Transformational similarity – however arbitrary the vocabulary used – is naturally a distance measure. Hence, it can be integrated in the world of quantitative measures. These should be merged with predicate-based measures. The best available approach appears to be the model introduced in the last section (Equation 28.17). Eventually, the powerful dual process model that covers the capabilities of qualitative, quantitative and transformational similarity can – if necessary – be embedded in a dynamic alignment process.

This unified model offers optimal human-like similarity measurement. Hence, it should support the representation of semantics and of polysemy as good as possible. However, this field of research is an active frontier in psychology as it is in computer science. The description of the state-of-the-art would require a book of its own. We are positive, though, that the future will see no fundamentally new way of similarity measurement. Rather, parameters and configurations of the dual process model will be improved, transformational similarity might become practically usable and the integration of the various parts in one model – as sketched above – will happen.

Chapter 29

Neural Media Understanding

Analyzes the building blocks of human cognition, explains how these are imitated in artificial neural networks and discusses practical networks for description, filtering and categorization, including the spike response mode, radial basis function networks and cascade correlation.

29.1 Neural Foundations

Man is the measure in media understanding. Therefore, the last step in this introduction must lead back to man. We started this book with the requirements of media understanding, introduced a large number of methods and explained here and there further ingredients of human-centered media analysis. Machine understanding is making progress, methods become more sophisticated and the results are – in selected domains – already quite acceptable. However, man is better, much better. Hence, the final question must be: How do we do it? What do the neural networks look like that perform sensual analysis in the central nervous system? *How can we imitate the human neural solution?*

The purpose of this chapter is to provide a first answer to these questions. As we will see in this section, human knowledge of the brain is still limited. Cognitive science is an active research frontier. Its results are of highest significance for media understanding – as for many other domains –, because eventually, our methods can only be successful if they imitate human behavior. Therefore, in this first section we summarize major findings about natural neural networks.

The second section explains the state-of-the-art of artificial neural networks and discusses the limitations compared to natural nets. The last two sections deal with (potential) artificial neural networks for description, filtering and categorization. Along the main theme, we introduce relevant neural network techniques such as radial basis function networks, cascade correlation and the spike response model.

The remainder of this section is structured as follows. We start with a short description of the neuron – the basic building block of the nervous system. Then, we discuss communication principles of neurons and, eventually, we briefly investigate the different types of memories, retrieval mechanisms and temporal firing patterns.

The human brain was already described in the first section of Chapter 23. The enormous number of 10^{10} neurons is interconnected by 10^{13} connections. Considering the size of the human brain, we have an average density of 10^5 neurons per cm^3. Major nervous centers are the *central nervous system* (CNS) and the *peripheral nervous system*. In the latter, the *autonomic system* controls heartbeat, respiration, etc. The *somatic system* is, for example, responsible for the control of body movement. The main task of the CNS is information processing: sensual perception, pattern recognition, storage, retrieval and similarity measurement.

Cognitive science could verify that the major functions of the CNS are organized in layers – for examples the centers described in Figure 23.1. Centers are linked by broadband connections. The majority of the neural cells are either responsible for sensual processing or motor control. *Afferent cells* propagate sensual stimuli to the CNS where *interneurons* provide information processing which causes *efferent cells* to trigger so-called effector cells attached to muscles. It is interesting to note that most mammals require the majority of neurons for motor control of large muscles (e.g. elephants). Furthermore, it could be shown that the feedback provided by motor cells is an important input for the evolution of the brain.

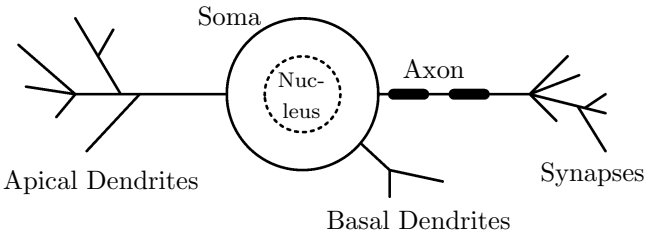


Figure 29.1: Structure of the Neuron.

Figure 29.1 illustrates the structure of the typical interneuron. The in-

put (left) is a tree of *dendrites* (branches), the output a corresponding tree of *synapses*. The *soma* is the central element with the *nucleus* at the core and the *axon* as its endpoint. In the neuron, information is propagated as an electrical potential. Between neurons (from synapse to dendrite) chemical *neurotransmitters* are used (mostly, glutamate). The usage of neurotransmitters causes two notable effects.

1. The speed of signal propagation is slowed down to about ten meters per second.

2. The output signal is weighted by a factor w: typically $0 < w < 1$ but it may also be $w > 1$ (*excitatory*) or even $w < 0$ (*inhibitory*). In the latter case, the output neuron's influence will be reversed.

Each neuron is connected to about three per cent of its neighbors in a diameter of one square millimeter. The average weight is only 1-5% of the input. The weight of a synapse depends generally on the distance from the axon hill. The nearer, the higher.

As we mentioned in Chapter 23, the typical *action potential* of the firing neuron lies at 50-80mV. The soma is able to hold the electrical potential of the input dendrites for a few milliseconds. Neural activity is limited to about two milliseconds. Neurons can be distinguished by the activity patterns they produce.

- *Phasic* neurons produce large bursts with low frequency.

- *Tonic* neurons produce low bursts with high frequency.

- So-called *fast spiking* neurons produce an intense *spike train*, i.e. combine phasic and tonic behavior.

The latter type of neuron appears with significantly lower frequency than the first two. Generally, the spike train produced by a neuron seams to be a diagnostic description of the information it processes. Hence, artificial neural networks of the latest generation take this type of information into account. We will discuss this issue in the third section of this chapter.

Eventually, we would like to point out that neural networks are based on the *all-or-none principle*. That is, a neuron will only fire, if all inputs together exceed at a particular point of time the threshold. Otherwise, the inputs are lost. This implies the need for recurrent structures and short-time memory in neural networks as a buffer for synchronization.

Neural theory distinguishes three types of neural memories. *Sensory memory* holds the information only for the first few milliseconds. *Short-time memory* is able to hold the input for a few seconds. Every bit of information that exists

longer in the CNS is held in the *long-time memory*. It appears probable that long-time memory is implemented in the form of weighted connections while the other forms could be (recurrent) propagation paths.

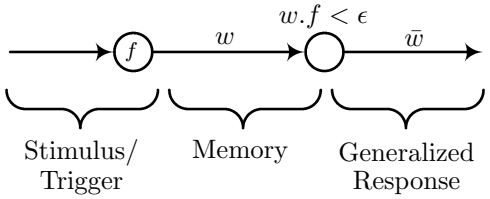


Figure 29.2: Generalizing Long-Time Memory Principle.

Figure 29.2 illustrates a long-time memory that is based on the *squared optimization criterion*. The memory is represented by the weight vector w that connects the input f (e.g. a query) with the generalized response $\bar{w}$. If the threshold ϵ of the memory neuron (the second one) is set $\epsilon \rightarrow w^2$ then only stimuli $f \rightarrow w$ will be able to trigger a response. Hence, only stimuli f similar to the 'hardcoded' weights w will be able to retrieve *generalized responses* (norms) from the memory. This model can easily be constructed by input patterns in a short-time memory and easily be extended to arbitrary memory sizes by parallelization. We believe that this model is a nice explanation of long-time memory and the generalization principle. It would also explain the near constant growth of connections (w) in the CNS throughout life.

Eventually, it appears natural that *neural processing over time* is based on similarity measurement (convolution). As we mentioned in earlier chapters, the *change of aspect* (oscillation in sign recognition) described by Wittgenstein and others can be described by a race situation in which two long-time memory patterns are alternatively triggered by the input stimulus. The oscillation could be caused by flickering perceptual stimuli that cause almost identical squared similarities in the described long-time memory. In consequence, the perception of the stimulus would oscillate. As a consequence, the temporal pattern (spike train, a form of attractor) that is produced by an input appears to be an interesting description of the recognition process.

Conclusion: The central nervous system is highly complex in structure and parallel dynamics. The machines of today are clearly unable to imitate the full richness of this apparatus. Still, some central functions are already understood today. How these can be modeled in computers is the topic of the next section.

29.2 Artificial Neural Networks

It is not our ambition to provide a full introduction into the field of *artificial neural networks* (ANN) in this section. Rather, we would like to introduce all concepts required to understand the neural networks for description and categorization presented in the last two sections. These concepts are all based on and imitate particular characteristics of natural neural networks. The question: *what is essential in neural networks?* has been discussed intensively during the last decades. Over time, the models have grown more complex. This development is reflected below.

First, we describe the representation of the neuron in ANN, aspects of the architecture and some early network types that were of paramount importance in the evolution of ANN. Then, we discuss learning strategies and limitations of learning and categorization for particular network types. The two main concepts introduced below are the McCulloch-Pitts neuronal model and backpropagation learning.

Neural networks can be distinguished by a number of properties of which the following three appear most important to us.

- Direction of the flow of information in the network
- Layer structure and complexity
- Type of the employed stimulus response model

The direction of the flow of information determines the *network type*. Most ANN are either *feed-forward*, *feed-back* or *recurrent* networks. Of the latter type of network we already encountered examples in earlier chapters. The Hopfield network and the Boltzmann machine are recurrent networks, because information flows back and forth in them until a stable state is reached.

The self-organizing map introduced in Chapter 19 is an example for a feed-forward network. Both in learning and categorization a strict feed-forward strategy is employed. The response of the network on the input is in both cases not used to refine the network structure or the weights. In the next section, we will briefly discuss the neural structure of the self-organizing map.

The majority of ANN, however, are feed-back networks. We already encountered the perceptron as one archetype of the support vector machine. Below, we discuss this type of network in detail, because it can be used as the foundation for media description and neural categorization networks.

ANN are typically differentiated by their layer structure into *single-layer* and *multi-layer* networks. In a single-layer network, the output layer is directly connected to the input layer (the input layer is not counted). Such networks are typically fully connected, i.e. every input node is connected to each output node. The self-organizing map is an example for this type of network.

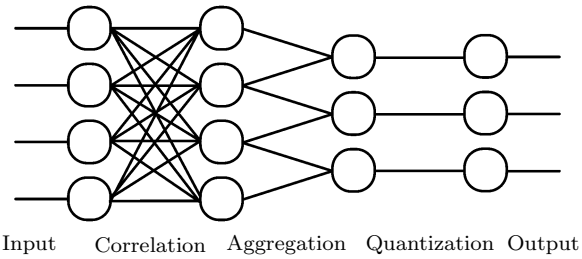


Figure 29.3: Multi-Layer Neural Network Example.

Figure 29.3 shows an example for a multi-layer network. The input layer and the output layer are separated by two *hidden layers*. As we can see, the complexity of the connections of the layers varies widely. While the input layer and the first hidden layer are fully connected (high complexity, high computational effort required), the two hidden layers are only connected by a pair-wise aggregation operation. Eventually, the second hidden layer and the output layer are linearly connected (low complexity). Semantically, *the full connection performs a correlation operation*. The two hidden layers aggregate the results locally. The linear connection performs just a quantization operation. These semantic meanings point already in the direction of the ideas developed in the next two sections: We will try to associate the building blocks of description and categorization with particular network types and connections.

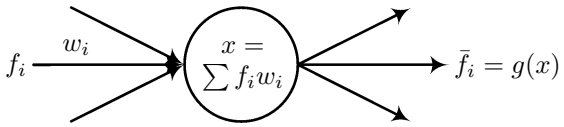


Figure 29.4: Artificial Neuron.

Figure 29.4 shows an artificial neuron – the basic building block of all ANN. The synapses and dendrites are represented by input values f_i and input weights w_i. The soma performs an aggregation operation of the weighted inputs. The nucleus contains an *activation function* g that transforms the input f into the output $\bar{f}$ in an all-or-nothing fashion. Three important activation functions are the step function (Equation 29.1), the sigmoid learning function (29.2) and the perceptron function (29.3).

$$g(x) = \begin{cases} 1 & \text{if } x \geq \epsilon \\ 0 & \text{else} \end{cases} \tag{29.1}$$

$$= \frac{1}{1 + e^{-ax}} \tag{29.2}$$

$$= \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{29.3}$$

Here, a is a scaling parameter and ϵ is the activation threshold. Based on these activation functions (responses), layer structures and directions of flow, we can define a number of classical ANN. The first is the *McCulloch-Pitts model*. It defines a two-layer network that employs a step activation function. This model can be used for simple categorization purposes (e.g. regression). However, its main value lies in being an early bird.

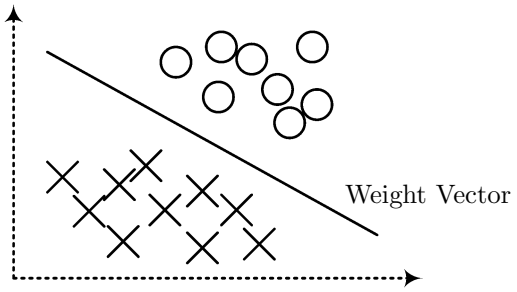


Figure 29.5: Perceptron Example.

Figure 29.5 illustrates the application of a perceptron neural network. The perceptron is a McCulloch-Pitts network that uses a feed-back algorithm for learning. Since the standard perceptron is not able to learn particular input-output patterns (XOR problem), the perceptron is sometimes extended by one hidden layer. The perceptron is practically usable: It provides a separating hyperplane (defined by the weight vector between input and output layer) that can be used for binary categorization and regression. As mentioned above, it was one of Vapnik's starting points in the development of the support vector machine, which is essentially a perceptron with a determined optimal orientation of the separating hyperplane.

Despite its generally interesting properties, the perceptron is hardly used in media understanding (and other domains) today. Next to the two named disadvantages (XOR learning problem, suboptimal goal definition) the perceptron is not able to learn a good separator for overlapping data. The simple layer structure does not allow for the definition of elastic mechanisms like slack variables.

Hence, feed-back networks with three or four layers are more frequently used in practice today.

For learning in ANN we have two fundamental options: *supervised* or *unsupervised*. The latter type is employed in feed-forward networks. There, the optimization criterion has to be given implicitly in the learning algorithm. For example, in the self-organizing map it is given as minimizing the quantization error by adaptation of the codebook vectors.

Feed-back networks and recurrent networks usually employ supervised learning. We have already sketched the simple learning procedure of the Hopfield network and the more sophisticated one of the Boltzmann machine. Generally, three supervised learning types are practically relevant.

1. *Propagation*: The new weights w are set by a linear function l of the error $\bar{f} - f_{gt}$ and a learning rate a. Here, f_{gt} is the output that should according to the ground truth (supervisor) be produced for input f. That is, $w = a.l(\bar{f} - f_{gt})$. Through the learning rate a the error is propagated from the output back through the network.

2. *Gradient ascent* uses a model similar to propagation, but l is a smooth function and $\frac{\partial l}{\partial w}$ is employed to reach the optimal weights quickly.

3. *Hebb learning* is similar to gradient ascent. The correspondence of inputs and outputs is reached by setting $w = a.l(f_i.f_j)$. That is, we use the covariance (outer product) of the input vectors to define their weights. Of course, this method works only if there is similarity (correlation) between the input and output patterns.

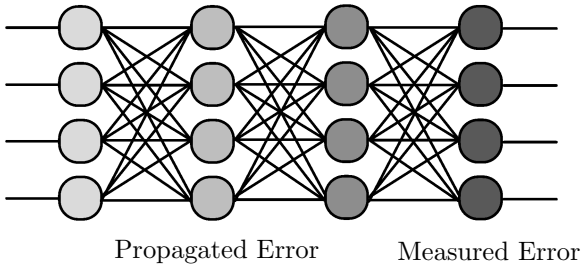


Figure 29.6: Principle of Backpropagation.

Supervised multi-layer perceptron-like networks usually employ the *backpropagation* algorithm for learning. It is based on the propagation principle and employs the following steps.

1. Select an input vector (randomly) from the training set and feed it into the network.

2. Compute the output and measure the error $\bar{f} - f_{gt}$.

3. If the error is above a pre-defined threshold, continue with the next step. Otherwise, stop learning.

4. Feed the error back into the network by adapting the last layer first by a fraction of the error, weighted by the learning rate.

5. Propagated the rest of the error back through the network. Figure 29.6 shows how this procedure distributes the refinement quantities over the entire network.

The backpropagation algorithm converges. Quantitative experiments conducted in the same fashion as those in Chapter 27 showed that there is no ground truth configuration that would cause oscillation or even chaotic behavior.

There are several ways to describe the capacity and complexity of an ANN. One is the layer structure. Another is the computation of the information-theoretic complexity in the form of the entropy. After training, the quality of the matching of input-output patterns can be estimated as $p(\bar{f}|f)$. These values can be employed to compute the conditional entropy of the network which is a measure for its complexity and generalization power. Furthermore, the VC dimension of ANN can be estimated as follows.

$$h_{ANN} \leq a.W. \log N \tag{29.4}$$

Here, W is the number of weights, N is the number of neurons in the model and $a \in [0, 1]$ is a contracting weight. That is, the capacity of an ANN will rise most effectively with more connections (like the human brain does) and only little with additional neurons.

In conclusion, there are various simplifications in artificial neural networks compared to natural ones. We can deal only with few simplified neurons, few layers, simplified learning procedures and simple application patterns. For this reason and because the effect of an ANN is often hard to analyze (if it works: why does it work?), they are despite their generally good categorization performance seldom used in media understanding applications.

However, neuralization must necessarily be the future research frontier of media understanding. Man can do it, man is the measure, therefore, science has to find a way. The next two sections are a first step in this direction: We introduce and suggest a number of networks for the representation of the fundamental media description and categorization problems.

29.3 Neural Description and Filtering

This section deals with neural description processes. We first introduce the spike response model as a general way of media description by neural processing. Then, we suggest neural equivalents for the building blocks of feature transformation. Eventually, we discuss an ANN for information filtering that extends the capabilities of the Hopfield network: adaptive resonance theory.

ANN are frequently used for categorization but hardly ever for feature transformation/summarization. The first two generations of neural networks considered only neuronal aspects (activation function) and synaptic aspects (weighting). With these tools it is possible to implement effective regressors. For summarization, however, a description of the (temporal) process is also required. This – statistical – view is provided by spiking neural networks. Hence, we describe this type of network in this section.

The mental representations of media objects in the human brain are constructed by neural processes. Stimuli are transformed to signs, statistically aggregated to norms and memorized as references. In the visual domain, for example, a sequence of saccadic scanning, color, depth and form detection (e.g. by on-off ganglia cells) represents the first steps of analysis. Later, low-level descriptions are grouped and organized, matched with memories of previous sensations and eventually laid down in a semantic storage that is enriched with language information and emotions.

A first, very general approach to imitate this behavior of the human brain is based on the statistical description of temporal recognition processes in the neural network. The idea is simple. Instead of considering the output of a multi-layer network, we focus on the *characteristics of the activation functions*. The *bursts over time* (*spikes, spike train*) caused by some input stimulus may be adequate descriptions of the input object, if the network performs reasonable summarization operations. Hence, we require two things: ANN components for summarization and an ANN model that provides the description. The remainder of this section aims at providing these components.

The *spike response model* (SRM) is an early approach for the description of neural behavior. The spike train f_i of one neuron is defined as follows.

$$f_i = \{t_i | g_i(t) \geq \epsilon\} \tag{29.5}$$

We consider the spike train to be a description of the input data. It consists of all times t_i when the aggregated input of the neuron g_i triggered a burst. ϵ is the threshold of the employed step function. The state g of neuron i at time t is determined as follows.

$$g_i(t) = \sum_{t_i \in f_i} a_1(t - t_i) + \sum_j \sum_{t_j} w_{ij} a_2(t - t_j) \tag{29.6}$$

The state (membrane potential) is determined by two components (terms). The first summarizes all earlier bursts and weights them with respect to the time passed since the burst. The left part of Figure 29.7 describes function a_1. The second term summarizes over all inputs j of neuron i. For these, all earlier bursts are summed up and downweighted by function a_2 (right part of the figure). The weights w_{ij} represent the synaptic behavior.

The state equation is used to determine the state of each neuron at each relevant point of time. The result of computation is a set of spikes for each neuron. The superset over all neurons describes the input media object.

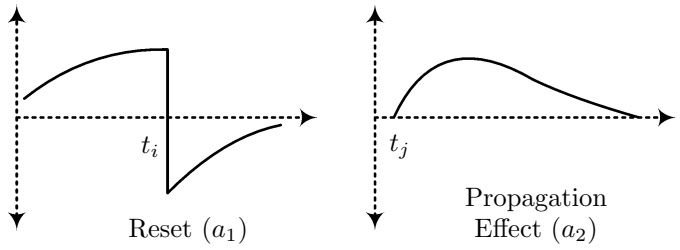


Figure 29.7: Spike Response Model Functions.

The second ingredient required for a neural description network is a neuralization of the building blocks of feature transformation. Once provided, complex feature transformations can be neuralized by simple recombination of the building blocks on different scales. The components suggested in the next paragraphs are based on signal propagation as explained in the second section. Where necessary, we use *inhibition* (that is, negative weighting). This function is reached in the human brain by the usage of a special neurotransmitter. In the ANN it is sufficient to use $-w$ instead of a normal weight.

In Chapter 11, we defined four building blocks of feature transformation: localization, interpretation, reduction (quantization) and aggregation. Of these, localization is a natural property of neural networks. Neural structures can process the input patterns in no other way than local and parallel. Examples for quantization and aggregation have already been provided: Figure 29.3 shows examples for both functions.

The only non-trivial building block is interpretation (cross-/autocorrelation). For this building block, Figure 29.8 shows a neural equivalent. The two input stimuli (of which one may be a template from memory or a piece of the same signal that provides the first stimulus) are processed in two steps. In the first, corresponding inputs are aggregated. In the second step, the locally aggregated (convoluted) components are globally aggregated. The design of the convolution

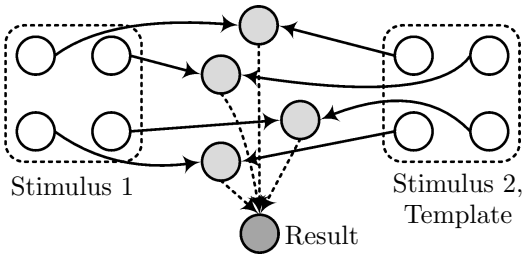


Figure 29.8: Neural Interpretation/Convolution Pattern.

operator is discussed in the next section as it is also the equivalent for the similarity measurement building block of categorization.

These building blocks can be used to construct neural representations of the major feature transformations. For the perception of visual stimuli, these should include color and form descriptions, relationships of symbols, movement paths and depth perception. As Kandel and Wurtz wrote in [186], these are also the major stimuli processed by the human brain (blobs, stripes, interblobs, interstripes). In the audio domain, the building blocks can be used to represent loudness and pitch perception, rhythm and timbre but as well advanced psychophysical concepts such sensory pleasantness. Of course, the same idea could also be applied to artificial stimuli that are processed by the brain only in the form of images.

We would like to close this section with a short look on neural methods for information filtering. In the second part, we already encountered dynamic quantization methods such as the Hopfield network and the Boltzmann machine. Formally, these methods are recurrent networks. A more sophisticated neural network for general information filtering is provided by the *adaptive resonance theory* (ART). This theory serves two purposes. On the one hand it describes human behavior in pattern learning and generalization and on the other it provides a practically usable model for convergent filtering.

The fundamental structure of ART is illustrated in Figure 29.9. The network consists of two layers (traditionally, named F1 and F2) which are fully connected. Information flows back and forth between the layers. Hence, ART is a recurrent network. The learning/categorization process of an input pattern consists of two steps. In the first, an input signal is provided through F1 and propagated by weights W to the output layer. In the output layer, the neuron with the maximal membrane potential is considered the winning node. All others are set to zero. In the second step, the signal is propagated back by weights Z to F1 and this resonance is used to measure the quality of the representation process. If the difference of the original input and the resonance is beyond a predefined

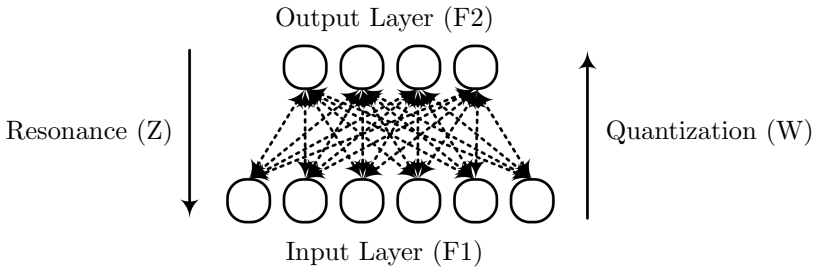


Figure 29.9: Adaptive Resonance Theory Network.

threshold, the process is repeated and the second-best neuron in F2 is chosen as the winning node, and so on. If no suitable neuron in F2 can be identified, a free neuron is chosen and the weights W, Z are set according to the input pattern. In this respect, the ART implements the generalized long-time memory model introduced in the first section.

The adaptive resonance theory is an interesting information filtering model. The two-layer structure allows for more efficient convergent filtering than in the Hopfield network. Several variants of ART exist that provide different functionality. We believe that this model could very well be used for information filtering in media understanding.

29.4 Neural Networks for Categorization

In this last section, we return to the 'natural' application of neural networks: categorization, the semantic interpretation of input patterns. First we review the neural definition of the self-organizing map of which the algorithm was already introduced in the second part. Then, we introduce radial basis function networks as typical static ANN for categorization and cascade correlation as an example for a dynamic ANN. Eventually, we do for categorization what we did for feature transformation in the last section: we define the building blocks – in particular, similarity measurement – neurally. On this ground, arbitrary machine learning methods can be assembled from the neural building blocks.

Figure 29.10 illustrates the neural network of the self-organizing map. It is a fully connected single-layer feed-forward network. The output classes c_j are, as mentioned in Chapter 19, organized in a rectangular or hexagonal grid. The weights w_{ij} that end in one class c_j define one codebook vector. Learning and application are defined as described in the second part.

A more complex neural network for categorization is illustrated in Figure 29.11. The *Radial basis function* (RBF) network is related to the self-organizing

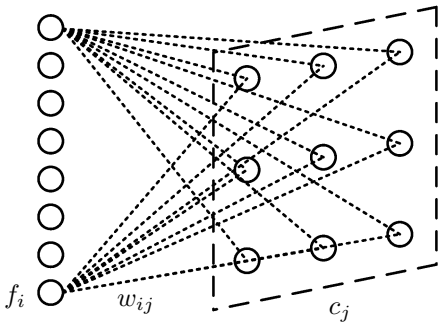


Figure 29.10: Self-Organizing Map Neural Network.

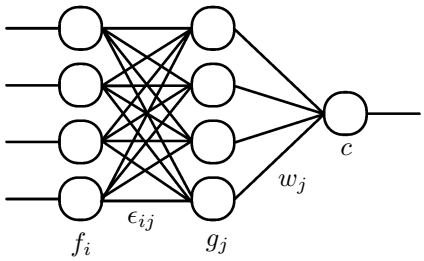


Figure 29.11: Radial Basis Functions Network.

map as the first part of the network has the same structure and serves the same purpose. The hidden layer and the output layer implement an aggregation pattern. The output of an RBF network is computed by the following rule.

$$c = \sum_j w_j g_j\big(m(f, \epsilon_j)\big) \tag{29.7}$$

That is, the class c is the neural sum of all codebook vectors represented by the so-called *centers* ϵ_j with weights w_j. The input f is compared (distance m) to all centers – as in the self-organizing map by the Euclidean distance. However, different to this network, in the second step not the node with minimal distance is considered the winning node. Instead, all centers contribute to the final classification. In this respect, the RBF network is similar to boosting. The activation function is usually a Gaussian generalization function.

$$g_j(x) = e^{-\frac{x^2}{2{\sigma_j}^2}} \tag{29.8}$$

Sometimes, the method of inverse multiquadrates with a free parameter a is used instead.

$$g_j(x) = \frac{1}{\sqrt{x^2 + a}} \tag{29.9}$$

As the authors of [171] point out, the RBF network is a nice example for a machine learning method that is heavily influenced by the results of psychological similarity research. The activation function is similar to the one suggested by Shepard for generalization. The association of input vectors and centers is based on Minkowski distances. There is also a link to the theory of ergodic systems. The RBF network defines an algebra over the input space f. The *receptive field* of center ϵ_i is defined as $\{f|g_i(m(f, \epsilon_i)) \geq a\}$. It is, therefore, not surprising that RBF networks are, for example, used to learn chaotic time series such as those produced by the logistic map.

Initialization of an RBF network based on ground truth is a four step process.

1. Select the type of the activation function g.
2. Select centers ϵ_j from the samples (e.g. mean values over groups).
3. If the activation function is Gaussian, set the standard deviation (*spread parameter*) as the second order moment (variance) of the vectors that contribute to one center vector.
4. Set the w_i so that they minimize the squared differences of centers and ground truth.

Additionally, the center vectors and weights can be refined iteratively (e.g. by expectation maximization).

Self-organizing map and radial basis function network are – as all other ANN considered so far – static methods. There are, however, also dynamic methods in which the network structure is adapted during learning (over time). One example that is of particular appeal for categorization is *cascade correlation* (CC). This method shows striking parallels to boosting – with all advantages and disadvantages.

Generally, a CC network is a fully connected multi-layer feed-forward network. Figure 29.12 gives examples. Goal is perfect categorization of input vectors f into classes c, formally:

$$\sum_j m\big(gt(f_j), c_j\big) \rightarrow \min \tag{29.10}$$

Please note that here j iterates over the training set. Over all input vectors the Euclidean distances m from the ground truth values gt to the class labels c of the winning node computed by the CC network should be minimal (least squared error). The learning process that should achieve this goal takes the following steps.

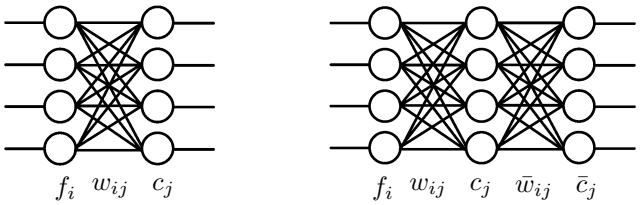


Figure 29.12: Cascade Correlation Example: Initial State (left), after one Iteration (right).

1. Find weights w_{ij} that optimize the criterion given above. The classes are defined as $c_j = \sum_i w_{ij} f_i$ for all input vectors. Here, i iterates over the description elements. This optimization is usually achieved by random initialization and expectation maximization. The result is equivalent to the left part of Figure 29.12.

2. If the squared error is beyond a predefined threshold, add a hidden layer of weights $\bar{w}$ with $\bar{c}_j = \sum_i \bar{w}_{ij} c_i$ that minimizes the squared error even better. Again, the $\bar{w}$ are identified by expectation maximization. The result is equivalent to the right part of Figure 29.12.

3. If the error is still beyond the threshold, return to Step 2.

CC networks are able to model any input data. The price is high computational complexity (expectation maximization for each new fully connected layer) and a high risk of overfitting. Still, CC networks are practically used, for example, for the learning of patterns and templates.

We would like to close this section with suggestions on how categorization methods can be neuralized. As for feature transformations, we suggest neural models for the building blocks of categorization. Concrete neural classifiers can then be assembled from the building blocks.

The four building blocks are quantization, estimation, learning and similarity measurement. Quantization has already been discussed in the previous section. Estimation for model building is a combination of quantization and aggregation – of course, similar to media description. Learning and refinement can best be modeled by recurrent patterns. In the second part, we pointed out that recurrent networks such as the Hopfield network are structurally identical to Markov processes. Hence, it appears reasonable to use these neural patterns for the representation of learning patterns.

The only building block that requires a sophisticated neural representation is similarity measurement. In the last chapter, we concluded that the currently

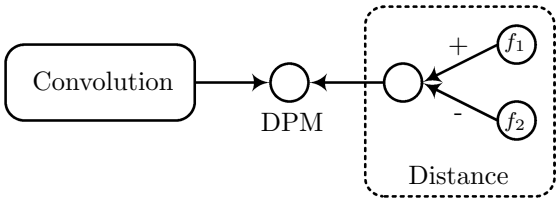


Figure 29.13: Neural Dual Process Model.

best similarity measure is a dual process model (DPM) of taxonomic an thematic thinking. Hence, we require positive and negative convolution for its representation. Figure 29.13 illustrates a straightforward approach. The similarity measurement DPM is a two step neural pattern. In the first step, positive convolution (as described in the previous section) and distance measurement are performed separately. Distance measurement is reached by inhibition of the second stimulus f_2 by the first. In natural neural networks, this is reached by the selection of specific neurotransmitters. In ANN, this behavior can be imitated by negative weights. Eventually, the results of positive and negative convolution are merged – possibly after generalizing quantization of the distance score – in the second step. We are convinced that this pattern provides a satisfactory explanation of dual process similarity measures in neural networks.

In conclusion, we provide a number of practically applicable artificial neural networks for media description and categorization in this chapter. Moreover, we suggest neural equivalents for the building blocks of feature transformation and the classifiers. These patterns can be used to build neural equivalents of the media understanding methods introduced in this book.

ANN categorization methods belong to the group of separators (e.g. the perceptron). The general effect of neural categorization methods is positive where semantics are concerned. However, this is paid with bad performance due to inadequate computation architectures of the machines employed today. Another problem is the dimensionality of the required solutions as ANN methods open large spaces of potential parameters and values.

Still, we believe that neuralization will be one frontier of future media understanding research. Since we endeavor to imitate an essential cognitive process of man, it is self-evident that the methods should be imitated as well. We see the presented neural building blocks as a first step in this direction. However, the frontier of neural media understanding will probably not be pushed too far in the near future. The reason is technical: With the present computer architectures, neural networks can only be simulated very slowly. Algorithmic approaches are currently clearly superior.

This chapter closes the practical introduction to media understanding pro-

vided by this book. The last chapter reviews all introduced methods, sets them into context and derives the essential messages from the text. Eventually, we provide an outlook on the media understanding challenges of the near future.

Chapter 30

Finale and Future

Summarizes the findings of the book, emphasizes the most important points, estimates the practical applicability of some important ideas and sketches a vision of future multimedia information retrieval research.

30.1 Summary

Like the Chapters 11 and 21, this one is dedicated to summarization and reflection. We repeat the most important findings – this time over all three parts – and endeavor to set them into context. The organization of all four sections follows the big picture, i.e. first come media-related issues, then feature transformations, information filtering and, eventually, categorization and evaluation.

This first section is a plain summary. The essential methods are listed and references to earlier chapters are given. In the second section, we gather the conclusions that are of paramount importance for practical multimedia information retrieval. The third section looks critically on the state-of-the-art of media understanding. Furthermore, we investigate the potentials of currently developed methods. In the last section, we give an outlook on what will happen next in media retrieval, what is likely to happen and what is rather improbable.

As one major asset of this first section, we provide three lists of methods that have high relevance in practical media understanding. The lists are more than just indices. We set terms into a hierarchical context and provide links to the chapters where they were primarily discussed.

On the general level, we have endeavored to develop a uniform theory of multimedia information retrieval that works well for media types as different as

video, text and bioinformation. Major components of this theory are the bigger picture of media understanding, a list of major interpretation problems, a common notion of feature spaces and a unifying notation for data and operations. The bottom line of the bigger picture is that media understanding is a three step process of summarization (feature transformation), redundancy elimination (filtering) and contextualization (categorization). The interpretation problems range from the psychological ability to represent semantics to technical problems such as computational performance. The definition of a common feature space requires most of all an understanding of scales, spaces and of measurement therein.

Feature Transformation: Essential Methods

- Feature Spaces (Chapter 7)
- Cognition, Perception, Psychophysics (Chapter 23)
- Transformations (Chapter 12)
- Temporal Media Description
  - Amplitude Summarization
    - Short-Time Energy (Chapter 4)
    - Sone Features (Chapter 13)
  - Frequency Summarization
    - Zero Crossings Rate (Chapter 4)
    - Mel Frequency Cepstral Coefficients (Chapter 13)
  - Autocorrelation
    - Correlogram (Chapter 4)
    - Cross-Spectral Density (Chapter 4)
    - Linear Prediction (Chapter 4)
    - Perceptual Linear Prediction (Chapter 13)
  - Template Matching (Chapter 24)
- Spatial Media Description
  - Color Description
    - Dominant Colors (Chapter 5)
    - Color Histogram (Chapter 5)
  - Texture Description
    - Moments (Chapter 5)
    - Multi-Resolution Analysis (Chapter 12)
  - Form Description
    - Edges (Chapter 5)
    - Local Description
      - Visual Keywords (Chapter 5)
      - Scale Spaces (Chapter 14)
      - Interest Point Detection (Chapter 14)
      - Gradient-Based Description (Chapter 14)
    - Templates
      - Active Contours (Chapter 24)
      - Curvature Scale Spaces (Chapter 24)
- Spatiotemporal Media Description
  - Temporal Segmentation (Chapter 15)
  - Optical Flow
    - Motion Activity (Chapter 15)
    - Object Motion, Motion Trajectories (Chapter 15)
    - Camera Motion (Chapter 15)
- Symbolic Media Description
  - Summarization (Chapter 6)
  - Histogram Building, Bags of Words (Chapter 6)
  - Structural Alignment (Chapter 8)

A second general ingredient of media understanding is knowledge about human perception and cognition. Though not all media (senses) considered in this book are natural (e.g. text), human perception is of highest interest for us, because we aim at imitating its way of data processing. Major topics discussed in this area include the physiology of the brain, processing paths in vision and hearing, norm theory and psychophysics (Mel scale, Bark scale, Steven's exponent, etc.).

We have explained a multitude of feature transformation methods that serve the purpose of media summarization. For temporal media (e.g. audio, biosignals, stock data) we have suggested summarization methods (e.g. short-time energy, zero crossings, mel frequency cepstral coefficients) and autocorrelation methods (linear predictive coding, correlogram, etc.). Template matching is also used (e.g. in technical chart analysis).

For spatial media (images), we have suggested methods for color summarization (dominant colors, color histograms), descriptions of surfaces (texture moments), of forms (active contours, shape moments, etc.) and of super-local components. In the latter area, interest point detectors are state-of-the-art. Their description by gradient neighborhoods and their aggregation in bags of features has been discussed in the second part of the book.

Video is the only spatiotemporal data type considered in multimedia information retrieval. We have encountered methods for scene segmentation (e.g. twin comparison). The essential description of motion is the optical flow. Of the several approaches that exist for flow computation, we have recommended the Lucas-Kanade approach. It can be used to measure the amount of object movement, movement paths (motion trajectories) and global motion such as camera motion.

For the summarization of symbolic media, we have suggested lossy compression, histogram-based methods (n-grams) and several structural alignment procedures with coarse representations.

Information Filtering: Essential Methods

- Description Fusion (Chapter 7)
- Redundancy Elimination
  - Principal Components Analysis (Chapter 7)
  - Singular Value Decomposition (Chapter 16)
  - Isomap (Chapter 16)
  - Feature Selection (Chapter 16)
- Quantization
  - Normalization (Chapter 7)
  - Kalman Filtering (Chapter 26)
  - Linear Vector Quantization (Chapter 26)
  - Hopfield Network (Chapter 26)

The information filtering methods discussed in this book serve three pur-

poses: merging of media descriptions, elimination of redundancies and quantization of description elements. In the first area, we have encountered static merging as well as early, late and hybrid fusion. The major method for redundancy elimination is factor analysis. Singular value decomposition and the Isomap approach are two related methods. Simple feature selection serves a similar purpose. Quantization methods range from simple normalization to convergence filters such as the Kalman filter, vector quantization and recurrent neural networks.

Categorization: Essential Methods

- Learning Theory
  - Concept Theories (Chapter 17)
  - Convergence, Dynamical Behavior (Chapter 27)
  - Generalization, Semantic Scale (Chapter 27)
  - Training, Limits of Learning (Chapter 17)
  - Micro Processes and Macro Processes (Chapter 17)
- Macro Processes
  - Hedgers
    - Metric Approaches
      - Cluster Analysis (Chapter 8)
      - Vector Space Model (Chapter 8)
      - K-Nearest Neighbor (Chapter 8)
      - K-Means (Chapter 8)
      - Self-Organizing Map (Chapter 19)
    - Mixture Models, Norms (Chapter 19)
  - Separators
    - Decision Trees, Random Forests (Chapter 8)
    - Bayesian Networks
      - Bayes Classifier (Chapter 9)
      - Markov Processes (Chapter 9)
    - Kernel-Based Methods
      - Support Vector Machine (Chapter 18)
      - Linear Discriminant Analysis (Chapter 18)
    - Artificial Neural Networks
      - Perceptron (Chapter 29)
      - Radial Basis Function Network (Chapter 29)
      - Cascade Correlation (Chapter 29)
      - Spike Response Model (Chapter 29)
    - Ensemble Methods, Boosting (Chapter 19)
- Micro Processes
  - Convolution, Kernels (Chapter 18)
  - Predicate-Based Measures (Chapter 28)
  - Dual Process Models (Chapter 28)
  - Similarity Meta Models (Chapter 8)
- Evaluation
  - Cross Validation (Chapter 20)
  - Recall, Precision, F_1 Score (Chapter 10)
  - Interestingness Measures, Entropy (Chapter 20)
  - Receiver Operating Characteristic Curves (Chapter 20)

Almost anything that was ever developed in machine learning is used in multimedia information retrieval. That is why we have described a multitude of

methods. On the theoretical side, we have discussed the particulars of learning processes and of generalization. Concept theories have helped us to understand the principal solutions for contextualization. We have introduced several tools for the differentiation of classifiers: into separators and hedgers, micro processes and macro processes, rigid versus overfitting methods, etc. The limits of learning have been described in terms of VC theory, PAC theory and dynamical systems theory. We have clustered the fundamental types of micro processes, derived a portfolio of categorization methods and suggested best methods for standard applications.

We have laid particular weight on the explanation of context and semantic meaning. As we described in the second part, *semantic understanding is the consequence of setting media summaries into the context of an application.* Different types of context were discussed. Media theory and semiotics were used to analyze and understand the input media and to derive the (hidden) semantic messages conveyed by them.

On the macro level, we have endeavored to provide an exhaustive list of presently employed machine learning techniques. We discussed tree-based approaches such as decision trees, random forests and cluster analysis. The latter method leads the way to the distance-based methods. This bag of methods includes the vector space model, k-nearest neighbor categorization, k-means and the self-organizing map. The last classifier provides a bridge to the artificial neural networks, which include the perceptron, radial basis function networks, cascade correlation and the spike response model. From the world of probabilistic methods we have discussed Bayesian networks in general, Markov processes, the Bayes classifier and important related methods such as mixture models. Eventually, we have dealt with risk minimization methods such as the support vector machine, linear discriminant analysis and ensemble methods such as boosting.

On the micro level, we have investigated the *communalities and differences of positive and negative convolution.* The similarity theory of dual process models is built on this distinction. Convolution is the essential method for template matching and autocorrelation. Hence, it is important in feature transformation and categorization. Distance-based methods are used directly for categorization (e.g. k-nearest neighbor classifier) but as well in similarity meta models (earth mover's distance, Hausdorff distance, dynamic time warping, etc.). There is a fundamental analogy between similarity measures and kernels. An extended similarity measure will include methods for transformational similarity, structural alignment and predicate-based measurement.

For practical multimedia information retrieval, we have introduced a number of evaluation measures and tools. The king's path employs cross validation as the testing procedure. Recall, precision, F_1 scores, interestingness measures, information entropy and related measures can be used for the summarization of experiments. The gathered results can be displayed in the form of receiver

operating characteristic curves. Tools for application building include Weka (machine learning), Matlab (feature extraction, categorization) and R (information filtering).

Furthermore, we have reviewed our results in various ways. We have distilled lists of building blocks for feature transformation and categorization. These were represented by neural models in the last chapter. The general goals of good description and good categorization where discussed in the two review chapters. Eventually, we have named and discussed the major challenges of media understanding today and in the near future. Important points such as the merging of interest point detection with Gestalt theory, the definition of canonical sets of description methods and pushing the frontier of neuralization are discussed in the remaining sections.

30.2 Essential Findings

This section lists the most important conclusions that should be drawn from this book. Media-related findings, feature transformation and categorization are investigated in this order. In the latter field we distinguish between the micro level and the macro level of categorization, because the findings in these areas build coherent clusters that are significantly different from each other.

We have three major media-related messages: Man is the measure when it comes to what semantic context is; all media can be treated in the same way; the rules of perception and cognition have to be taken into account in the processing of media data. These findings are discussed in the consecutive paragraphs.

1. We have to *put the human in the loop of iterative media understanding* because *man is the measure*. We have introduced the *semantic scale* to raise the sensibility for this issue. Media understanding methods are often subsemantic (for example, most feature transformations) but as well often supersemantic (for example, metric distance measurement in categorization). Both of these traps should be avoided, because they lower the acceptance of media understanding results. Therefore, we have to consider the results of psychology, psychophysics and cognition in the design of multimedia information retrieval methods. These issues are discussed below. As long as computational multimedia information retrieval trails behind human sign recognition, it is advisable to integrate the user in the – then, semiautomatic – media understanding process.

2. The media considered in this book have properties that make them significantly different from each other. Next to varying numbers and types of dimensions we have the fundamental problem of quantitative and qualitative samples: numbers and symbols. Still, as we could show, *the basic*

scheme of media understanding remains the same: Summarization transforms the input media object into a vector of description elements. Information filtering can be used to improve the data quality of this description vector. Eventually, a categorization process is employed that sets the summary into the context of an application. That requires the formulation of a ground truth, i.e. the definition of signs, the frequency of their appearance and their numeric description. In one sentence, *the classifiers uses the ground truth to transform the topology of feature space according to the semantics of the application.* This scheme can be applied iteratively to improve once acquired predicates (class labels) further and thereby raise the semantic level of the application. We conclude that the different properties of the media are captured by the feature transformations. As soon as we have a description space, differences in the input media do not matter anymore.

3. *Psychophysics and other areas of psychological research provide us with numerous valuable insights on how humans perceive media objects.* It is essential for computational machine understanding to imitate the majority of these particularities. For example, in audio understanding we have to respect the differences between frequency and pitch perception as well as the difference between sound pressure level and perceived loudness. In the visual domain, we have to deal with the differences in the perception of particular colors. The recognition of edges and interest points is based on more than just rational criteria. Psychological rules such as the laws of Gestalt play an important role. Perceptual, cognitive and statistical illusions have to be considered as well. Even though the majority of these illusions stand for errors of cognition, these *errors are still typically human.* It would not make sense to take a supersemantic point of view and neglect these human insufficiencies – ignorance of multimedia information retrieval would be the results. In the contrary, we have to appreciate them. Norm theory is a field of particular interest, as issues such as representativeness, anchoring and others have unfortunately been widely neglected – even in areas where the methodology can be characterized as psychological (e.g. technical chart analysis).

In the areas of feature transformation and information filtering we have introduced dozens of methods that are applied in multimedia information retrieval today. We have endeavored to re-introduce once famous, now almost forgotten methods as well as those currently hyped that have potential for the future. In the area of information filtering, principal component analysis is the method of choice for redundancy elimination between the dimensions of feature space. The Kalman filter appears near-optimal for the quantization of individual description

elements. When it comes to feature transformation, we would like to point out three major conclusions.

1. However different the input media and the desired results of feature transformation, *there are only three principal types of descriptions: points, intervals and point sets*. Examples for point features are peaks, averages and other statistical moments. Typical intervals are variances and belief scores. Point sets include histograms, templates and other semantic signs. These three types of *descriptions can be computed by summarization and correlation*. One typical form of summarization is averaging. Peak detection is another. The two fundamental correlation operations are autocorrelation and template matching. Interestingly, summarization and correlation can be performed on symbols as well as quantities. Symbolic summarization is, for example, text summarization. Structural alignment procedures (e.g. dynamic time warping) provide a form of symbolic template matching. In the quantitative domain, histogram building and edge detection are typical examples.

2. Summarization and correlation appear in their clearest form in the feature transformations for one-dimensional (often, temporal) media. Two examples from the audio domain are the *mel frequency cepstral coefficients and linear predictive coding*. The first method provides a highly efficient summary of the input media. Hence, processing has to include windowing and decorrelation. Linear predictive coding is the classic autocorrelation method – important in audio understanding since recurring patterns are characteristic for this data type. Schemes similar to these two feature transformations can also be applied on the other one-dimensional data types. The correlogram is an autocorrelation method employed in biosignal understanding. The sliding average is a summarization method in the stock domain. The bottom line is that each feature transformation will provide one of two functions: summarization or correlation. A healthy media understanding process will include both types of feature transformations: often, first summarization and then sign recognition by correlation of the summary with a template.

3. The visual sense is of particular interest for multimedia information retrieval today. The digital culture of the 21st century is a visual one. Therefore, we have introduced a number of visual feature transformations. Like many others, we believe that *the visual description method of the future is interest point detection*. However, in order to be successful, the approach has to fulfill two requirements. Firstly, it has to take object boundaries into account. Secondly, it has to pay respect to the human way of perception (psychophysics, see above). The first requirement can partially be

fulfilled by the method directly. Interest points have a tendency to lie at object boundaries and to gather at locations of concentrated information. Additionally, the approach can be supported by semantic knowledge and image segmentation. The most important message from psychophysics is in our opinion that *redundancy matters* in the visual domain. Objects have to be described by more points than just the characteristic ones in order to be adequate. See the discussion in Chapter 14 for details.

Categorization and evaluation is the second major step of media understanding. This is the point where human judgment comes into play: most notably in the micro process but structurally as well in the macro process. Below, we first list essential findings about the micro process and then about the macro process of categorization.

1. *Machine learning reflects some results of learning theory and of the psychology of similarity, others not.* For example, generalization is employed in some micro processes and kernel functions but not in others. We could show that the number of fundamental micro processes is limited: rule-base, distance-based, belief-based and quantization-based are essentially the methods of choice. All of these processes represent some of the neural characteristics of human cognition – none all. We have argued that *the optimal micro process is probably a dual process model of positive and negative convolution.* Convolution is the operator that implements correlation. Frequently, positive convolution is employed for summarization while negative convolution is employed for autocorrelation and template matching. Intelligent kernel functions should behave like classifiers. In particular, the kernel function is a similarity measure. Hence, it has to take generalization and the thematic taxonomic configuration of the user into account.

2. We could show that *the two worlds of quantitative and qualitative similarity measurement can be joined.* In the past, distance measures have been employed to measure the similarity between quantities. Predicate-based measures have been used to compare symbols. Recent psychological findings suggest that both approaches should be integrated in a dual process model. In order to reach this goal, we have endeavored to define two scales of measures. We could show that the quantitative measures can be explained as combinations of the dot product and the first order Minkowski distance (city block distance). The predicative scale explains the predicate-based measures as combinations of the number of co-occurrences and the Hamming distance. In the next step, we showed the equivalence between dot product and the number of co-occurrences for interval-scaled and nominal-scaled data. In the same way, city block distance and Hamming distance

are equivalent. Eventually, we could associate the similarity measurement end of the joint scale (dot produce, number of co-occurrences) with positive convolution and the distance measurement end with negative convolution, which clears the way for a convolution-based dual micro process.

3. Psychology says that *every individual has a characteristic tendency to judge some similarity measurement problems thematically and others taxonomically.* The relevance of this issue becomes clear when we understand that taxonomic thinking is equivalent to positive convolution while negative convolution is the operator for thematic thinking. We conclude that for the successful dual process model we have to identify the configuration of the user. In the third part we have suggested a triad-based test for the recognition of this parameter. The test uses phenomenological stimuli as suggested by psychologists.

The major issues on the level of the macro process have to do with *making the large set of machine learning techniques handleable for media understanding* application.

1. Hence, we have defined a simple yet effective portfolio of classifiers. *The essential criterion for the macro process of categorization is whether it is separating or hedging.* A separating classifier will try to cut feature space in the right way in two halves. In contrast, a hedger will endeavor to fence off consistent regions in feature space and label them semantically. The two types of classifiers have a representation in concept philosophy. We could show that the classic theory is equivalent to separating while the prototype theory is equivalent to hedging. Since concept theory, i.e. the definition of signs by humans, has been discussed for more than 2400 years we can assume that all the fundamental answers are known today. This makes us confident that the pool of machine learning methods covers all relevant approaches as well. This insight reduces the machine learning step of multimedia information retrieval to the selection of an appropriate method. Then, the right mixture of flexibility and avoidance of overfitting depends essentially on the quality of the training data.

2. *Media understanding has to be employed iteratively in order to have a chance of success.* Above, we already stressed how important it is to put the human in the loop. Even if this is not possible, it makes sense to employ media understanding iteratively. In the first iteration, the quantitative summaries can be transformed to (proto-)predicates that have little semantic value. Consecutive rounds of media understanding will try to elevate the semantic level step by step. In consequence, it will make sense to employ different classifiers. In the earlier iterations, simple methods

such as decision trees will have a high chance of success. Later iterations will require the application of methods that are able to quantify the belief in the direction of improvement. As psychological findings show, human judgments vary widely and our confidence in our opinions is most of the time not very strong. These particulars can be represented in multimedia information retrieval for example by Bayesian methods.

3. Of the various evaluation methods introduced in this book, *the measure of stability and discrimination* that was defined in Chapter 20 *measures exactly the properties of good feature spaces.* Since the quality of categorization stands and falls with the topology of feature space, we strongly recommend using this measure to judge the quality of the feature transformations and – if necessary – exchange them for more discriminating methods.

The principal line of this book is to move from simple to complex problems. Understanding the solutions for simple problems allowed us to analyze the more complex problems. The result was a set of building blocks that are employed in the feature transformations and the categorization methods. Quantization and correlation/similarity measurement are the two methods that are central for multimedia information retrieval. The first requires understanding the input media, the second understanding the user.

30.3 Critical Review

This section should not be misunderstood as a general analysis of the feasibility of media understanding approaches. Hopefully, we have provided this analysis as a structural aspect of the entire discussion so far. Instead, we pick selected issues that may have been underrepresented or even neglected. The flow of argumentation follows – as always – the big picture.

In the introduction we argued that multimedia information retrieval is a combination of signal processing (in the widest sense) and machine learning. Looking back on the heap of methods this picture needs a slight correction. Media understanding is neither straight signal processing nor straight machine learning. For example, we do not require back transformations for the transforms used to escape the gravity of the sample. Furthermore, we do not require our classifiers to actually separate the data. This function should be provided by the discriminating features. Eventually, the two domains (feature extraction and categorization) merge more and more. As we saw in the discussion of the building blocks: some ingredients of feature transformation are highly similar to ingredients of categorization schemes. Quantization and correlation/similarity measurement are the two outstanding examples. The analysis of methods and

the neuralization of building blocks lead the way. We are positive that future media understanding research will see an amalgamation of the employed methods.

So far, however, even the results for well-defined semantic application domains are – to say the least – mediocre. The authors of [287] have investigated this phenomenon on a general level and have arrived at some very interesting conclusions. For the three principal dimensions: training data, feature transformations, classifiers they have compared machine performance to human performance. The result is that the training data and machine learning procedures of man and computer perform comparably. The major difference lies in the descriptions. The human brain appears to possess more flexible, detailed and discriminative sign identification procedures. In comparison, our computational feature transformations are slow, unnecessarily global, representative and non-discriminative. With the authors of this paper, we conclude that future multimedia information retrieval research will benefit most from investments in smarter feature transformations. The toolbox of categorization seems to be quite complete. The toolbox of media summarization is not.

In the feature extraction domain, we have argued in the second part that multi-dimensional wavelets would be desirable. The obvious advantage of such wavelets would be straightforward transformation of the input data into a semantically relevant spectrum. However, the disadvantages may also not be neglected. The major problem would be an explosion of the dimensionality problem. Necessarily, a multi-dimensional mother function would require several parameters and scales of values. If this scheme should work, we would require human-based research that tells us what parameters are really important and which values they realistically might have. As often, human experience tells us that we hardly ever have a balanced, complete view/model of our environment or some decision problem. Still, we are able to find orientation and act reasonably. The same philosophy would have to be introduced for successful media description: What parameters are essential? What values may really occur? We conclude that pushing the frontier of description would involve many more psychological user studies.

A minor question that we might ask ourselves is whether or not there is a gender aspect in the feature transformations employed today. The author noticed in audio-related work that some audio feature transformations are sexist in the sense that they are optimized for male speakers/singers. We believe that such a gender imbalance will almost never be the result of the processing scheme, rather of magic quantization. It goes without saying that such insufficiencies need to be eliminated where they exist and avoided through gender-balanced training media and ground truth. This is one reason why more weights should be laid on the quality of the training data in multimedia information retrieval.

On the categorization level, one issue of interest is whether or not in times

of dual process models the artful previous solutions for correct psychological similarity measurement still make sense. One is the Krumhansl model that was already discussed in the first part of the book. Another are psychological Minkowski distances (Equation Q1 in Appendix B.1) that add two degrees of freedom: Firstly, the a_2-th root of the a_1-th power of the differences is used. The difference $a_2 \neq a_1$ introduces several non-metric aspects of human-like measurement. Secondly, measures with $a_1 < 1$ are also non-metric. At least such a a_1 allows for violating the triangle equality. We believe that these models are now obsolete since the thematic taxonomic parameter of the suggested dual process model allows for defining arbitrary non-metric measures. That is, the ideal dual process model makes the majority of the measures listed in the Appendix obsolete. An exception are the dynamic association measures (similarity meta models) since these introduce an alignment meta process that is probably justified by human cognition/behavior.

After all our praising of dual process models it may appear heretical to ask if dual process models will really be the final solution for similarity measurement in categorization micro processes? Future will tell. We believe that the time for dual process models in machine learning will come. However, if their introduction is not accompanied by a proper test for the thematic taxonomic configuration of the individual user, the entire scheme may turn out as inflexible and inadequate as the models employed today.

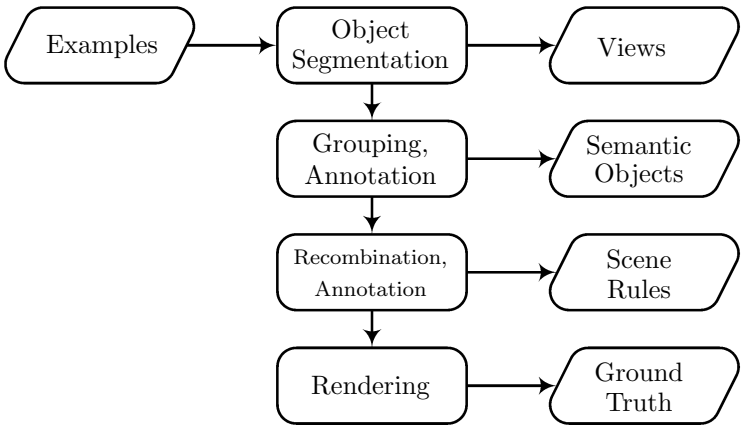


Figure 30.1: Fast Ground Truth Building Procedure.

One final issue that needs discussion is ground truth building. We have seen that ground truth is essential for many learning macro processes. We have emphasized several times that an appropriate *ground truth has to be well-balanced, complete, statistically representative, etc.* Such a ground truth is hard to obtain

– in particular, if there is no perspective of scientific merit involved. Furthermore, the methods required for ground truth building are clearly not engineering methods. Rather, psycho-/sociological know-how is required. Therefore and despite that, we would like to suggest a general scheme for efficient ground truth building (Figure 30.1). The idea is semiotic: What we want is not an assembly of media objects, but one of signs. Hence, in the first step we will take a set of media objects that contains all relevant signs in the major views and segment it by hand. The result is a large set of views of signs. In the second step, a statistically representative group of users (balanced gender, age groups, etc.) will annotate the signs semantically. The result is a list of semantic objects. In the next step, groups of signs are related by users. The goal is to produce as many meaningful combinations as possible. Relations are annotated as well, which allows for the derivation of a set of scene composition rules. With annotated views, composition rules and a statistical description of the user groups of the to be developed media understanding application it is possible to render an adequate ground truth automatically. We believe that this divide and conquer strategy allows for solving the ground truth problem in a way superior to the one used today.

The mentioned problems of multimedia information retrieval are only a few. Many more do exist and have been addressed in this book. In the final section we would like to turn our eyes from past and present to the future.

30.4 Outlook: To Do List

Now, we would like to take the opportunity and emphasize a few points that should be solved in the near future of multimedia information retrieval research. Other potential directions of future research were named and described in the text where appropriate. Hence, we do not provide a full *to do list* below but only one of selected issues that appear novel, interesting and rewarding to us. The organization is as always: first come the media, then the descriptions, and eventually the semantics.

The world of digital media is large: the retrieval problem can be approached over various paths, some of which where pursued in this book. Many strategic and operational issues remain unsolved today. One development that seems relevant to us is that through digitalization the technical media characteristics become more and more irrelevant. The shape of the digital media converges. Therefore, maybe the summarization methods should also converge.

1. We consider a number of heterogenous digital media in this book and show that they can be treated in very similar ways. We believe that the future input of media understanding will be an *unimedium* composed of an (almost) arbitrary number of channels with different origin and characteristics. On

this unimedium all principal summarization methods should be applied in order to provide the feature ground for semantic contextualization by categorization. One first step in this direction would be the application of audio feature transformations on other one-dimensional media types but as well on visual data and symbolic media. It may also be rewarding to apply spatial feature transformations on pseudo-spatial data such as windowed Fourier spectra. Further below, we suggest a general scheme for automatic description selection from such mixed feature transformations.

2. Successful multimedia information retrieval requires more specialized interactive applications. For example, a zoological application would be desirable that learns the rules of taxonomic description during application. This would require a learning algorithm flexible enough for exchanging description elements during training and application – like humans do [287].

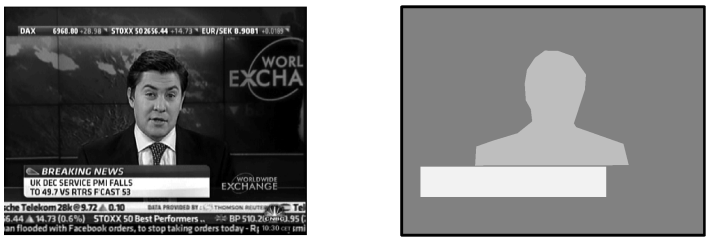


Figure 30.2: 3D TV Depth Map Example (© *CNBC*).

3. One practical stimulation of future visual media understanding is the introduction of 3D television. However depth information is introduced (e.g. as a depth map, see the right part of Figure 30.2), it can be employed to recognize object boundaries. Then, we will become able to apply local descriptions of objects with higher accurateness and chance of success. An early iteration of media understanding will lead to proto-signs that can be improved and related iteratively in refinement cycles. We firmly believe that the spreading of visual 3D content will be a great help for visual media understanding.

The selection of features in media understanding is – compared to humans – slow, inflexible and superficial [287]. More needs to be done in order to identify the *diagnostic features* as early as human perception and to use them as effectively as our cognition.

1. We suppose to apply all available summarization and correlation methods on the proposed unimedium. The result is necessarily an overcomplete description of the input media content. For the reduction of feature

space we suggest using the described information filtering methods but as well ground truth-based methods. For example, an optimization algorithm could be used to align the most discriminative description elements with ground truth values of training samples. Genetic algorithms are tailor-made for this purpose. For the evaluation, the measures introduced in Chapter 20 could be employed. In a similar fashion, artificial feature transformations could be rendered from the building blocks by genetic recombination and evaluation of their performance.

Figure 30.3: Local Description of Color Boundaries (© *CNBC*).

2. Interest points and flow vectors should only be computed at object boundaries and color segment boundaries. In the first part, we mentioned that the luminance values typically employed for grayscale representation in computer vision emphasize some colors while neglecting others. Hence, local features detected on gray ground are only partially similar to features detected on chromatic ground. Figure 30.3 shows an example. We suggest performing local feature detection on well-defined hue values instead of luminance values. Furthermore, the Gestalt laws should be taken into account in order to guarantee full human-like description of visual objects.

3. The last word has not been spoken on the feature selection issue. Humans select diagnostic features quickly, computers fail. One approach to improve the quality of descriptions could be an iterative media understanding scheme in which short descriptions (only the first factors) are employed which are enriched by additional description elements in consecutive iterations. That would limit the recognition problem in the first iterations and open additional space for discrimination in later – semantically higher – iterations.

Though the toolbox of categorization appears quite satisfactory, there are still a number of issues unsolved. Independent of the question if it would be desirable to define an 'ideal' classifier, work needs to be done to take over the latest results of psychological research and learning theory into machine learning.

1. An ideal classifier could look like a self-organizing map that employs the Tenenbaum function for generalization, the proposed dual process model for similarity measurement, factor analysis for redundancy elimination, a choice model for the selection of the winning node and the feature selection scheme introduced above. This classifier could be employed to provide an early orientation point for categorization. The resulting map would help to understand the topology of the media space and, in consequence, to select an appropriate classifier (automatically) for the actual application from the toolbox.

2. Following an idea of Kemp and Tenenbaum it would be desirable to take the *form* of description space into account in categorization. Here, we can understand form as the *best possible visualization of the data*. The Isomap approach is a first attempt in this direction. Conditional probabilities of the form $P(form|description)$ could be employed to refine the belief scores of classification. For example, a low conditional probability would reduce the belief score as the description would obviously not fit the form of the input data.

3. Eventually, we would like to emphasize once again that the findings of psychological similarity research need to be reflected in multimedia information retrieval. There is more than just the Euclidean distance: Dual process models with individual configuration represent human similarity judgment. Structural alignment as a meta-process and transformational similarity as a specific distance measure augment the process. Psychological similarity measurement should also be employed in kernel functions. Distance to similarity conversion needs to be based on proper generalization functions.

Our final word is that we hope that this introduction to multimedia information retrieval provides a fair ground for practical work. We are convinced that media understanding will gain more attention in the future. Methods should be exchanged between research disciplines. Man should be the measure in application design. *Goggles* is certainly not the last word.

Part IV

Appendices and Indices

Appendix A

Mathematical Notation

A.1 Sets and Arrays

Name	*Description*	*Type*	*Definition*	*Usage*
O	Medium (d-dimensional)	Array	$[s_l \mid s \in S \wedge l \in L^d]$	$o \in O$
F	Description (1-dim)	Array	$[s_l \mid s \in S \wedge l \in L]$	$f \in F$
C	Class (0-dim)	Array	$[s_l \mid s \in S \wedge l = \emptyset]$	$c \in C$
S	Sample	Set	$\{x_1, x_2, ..\}$ with x_i float	$s \in S$
L	Location	Set	$\{x_1, x_2, ..\}$ with x_i float	$l \in L^d$

Table A.1: Sets and Arrays.

A.2 Pre-defined Location Sets

Set	*Dimensions*	*Description*
L_{time}	{time}	Time (Audio, Biosignal, Stock)
L_{pos}	{x}	Position (Bioinformation, Text)
L_{point}	{x,y[,time]}	Point (Image, Video)
L_{moore}	{x,y}	Moore neighborhood (the eight cells surrounding a central cell on a square lattice)
L_{vn}	{x,y}	Von Neumann neighborhood (3x3 cross)

Table A.2: Important Media Templates.

A.3 Media Templates

Medium	*Description*
o_{bark}	Barkhausen curve
o_{cosine}	Cosine wave
o_{gauss}	Gaussian filter
o_{haar}	Haar wavelet
o_{lap}	Laplace operator
o_{mel}	Mel curve
o_{mexhat}	Mexican hat wave
o_{normal}	Normal distribution
o_{sine}	Sine wave
o_{sobh}	Horizontal Sobel operator
o_{sobv}	Vertical Sobel operator

Table A.3: Important Media Templates.

A.4 Variables

Value	*Category*	*Description*
$\mu, m, \sigma, v, \kappa$	Moment	Statistical mean, median, standard deviation, variance, skewness
$\mu_{k,r}$	Moment	Moment of k-th order at value r
λ, v	Moment	Eigenvalues and eigenvectors or singular values and singular vectors
χ_{xy}, ρ_{xy}	Moment	Covariance and correlation of objects x, y
b, e	Moment	Belief, criterion, entropy, energy
r, p, f_1	Moment	Recall, precision, F_1 score
a_i	Parameter	Free parameter
α_i	Parameter	Bound parameter
$\delta, \delta\delta$	Parameter	Change, change of change (e.g. in locations)
ϵ_i	Parameter	Limit, error or threshold
h	Parameter	Window size, VC dimension
w	Weight	Scalar, vector or array of weights

Table A.4: Important Values and Vectors.

A.5 Operations

Operation	*Description*
$\|.\|$	Absolute value
$\| \, . \, \|$	Normed dot product
$x + y$	Merging of objects x, y
$x \otimes y$	Convolution based on product
$x \bar{\otimes} y$	Convolution based on difference
$\theta(o, l, \epsilon)$	Neighborhood ϵ of object o at location l
$dims(o)$	Number of dimensions of object o
$size(o) = \|o\|$	Size of object
$y = cut(o, l_{start}, l_{end})$	Cut object y from object o from l_{start} to l_{end}
$y = chn(o, n)$	Cut n-th channel of object o into y
$round(x, y)$	Rounds x to y bits accuracy
$win(o)$	Window smoothing function
$gt(o)$	Ground truth value of object o
$m(x, y), m^{-1}(x, y)$	Similarity measure, distance measure
$m_n(x, y)$	Meta-similarity measure (e.g. n =Hausdorff)
$k(x, y)$	Kernel similarity measure
$mean(x), stddev(x)$	Statistical moments
$E(f)$	Expected value
$P(x = n)$	Probability of $x = n$
$P(x, y)$	Joint probability
$P(x\|y)$	Conditional probability
$Q(x)$	Mixture of probability functions
$perm(f)$	Power set
$pp(o)$	Psychophysical transformation
$y = ct(x), x = ct^{-1}(y)$	Cosine transformation
$y = ft(x), x = ft^{-1}(y)$	Fourier transformation
$y = wt(x), x = wt^{-1}(y)$	Wavelet transformation
$radon(o)$, $hough(f)$	Radon/Hough transformation

Table A.5: Important Operations.

A.6 Building Blocks

Operation	*Group*	*Description*
$acorr_i()$	Extraction	Autocorrelation
$agg_i()$	Extraction	Aggregation
$classify()$	Categorization	Mother function
$ccorr_i()$	Extraction	Crosscorrelation
$dcorr_i()$	Extraction	Interpretation
$estimate_i()$	Categorization	Density estimation
$filter()$	Filtering	Mother function
$learn_i()$	Categorization	Control loop
$loc_i()$	Extraction	Localization
$measure_i()$	Categorization	Similarity measurement
$quant_i()$	General	Quantization
$transform()$	Extraction	Mother function

Table A.6: Building Blocks.

A.7 Pseudo-Code Format

```
if .. then .. elsif ... else .. endif
for .. do .. endfor
foreach .. in .. do .. endfor
do .. while .. enddo
while .. do .. enddo
function .. takes .. begin .. return .. end
```

A.8 Some Expressions

Element	*Expression*
Assignment	$:=$
Equality	$=$
Range of values	$begin : end[: step]$
Standard iterators	i, j
Standard variables	x, y, z

Table A.7: Important expressions.

Appendix B

Similarity Models

See Chapter 28 for a thorough discussion of the listed measures.

B.1 Quantitative Similarity Measures

Below, $x, y \in F$ are descriptions with elements x_i, y_i, $K = size(x) = size(y)$.

Table B.1: Catalogue of Quantitative Similarity Measures.

No.	*Measure*	*Description*
Q1	$\sqrt[a_2]{\frac{\sum_i \lvert x_i - y_i \rvert^{a_1}}{K}}$	Generalized Minkowski distance group of parameters a_1, a_2 and dimensionality K (e.g. [215]). For $a_1 = a_2 = 1, 2, \infty$ we receive city block distance, Euclidean distance and Chebyshev distance, respectively. The latter is defined as $\max_i \lvert x_i - y_i \rvert$.
Q2	$\frac{\sum_i x_i y_i}{K}$	Dot product (e.g. [337])
Q3	$\frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$	Cosine measure, e.g., Gower 1967 [136]
Q4	$\sum_i \sum_j (x_i - y_i)(x_j - y_j)$	Mahalanobis distance group (with uniform covariance weights) [239]
Q5	$\sum_i x_i . log \frac{x_i}{y_i}$	Kullback-Leibler divergence [218]
Q6	$-\sum_i log\sqrt{x_i y_i}$	Bhattacharyya distance [32]
Q7	$\sqrt{2 - \frac{2}{\sum_i log\sqrt{x_i y_i}}}$	Hellinger distance [404]

...continued on next page

Table B.1: Catalogue of Quantitative Similarity Measures.

No.	*Measure*	*Description*
Q8	$\sum_i \frac{x_i - y_i}{x_i + y_i}$	Canberra metric, Lance and Williams 1967 [219]. Also known as χ^2 distance [315].
Q9	$\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sum_i y_i^2 - \sum_i x_i y_i}$	Tanimoto index [365]
Q10	$\frac{\sum_i \frac{(x_i - y_i)^2}{x_i + y_i}}{K}$	Divergence coefficient, Clark 1952 [55]
Q11	$\frac{\sum_i \frac{(x_i - \mu)(y_i + \mu)}{\sigma}}{K}$	Intra-class coefficient for classes described by μ, σ, Webster 1952 [393]. For $\mu = 0, \sigma = 1$ equivalent to the dot product.
Q12	$\frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_i (x_i - \mu_x)^2}\sqrt{\sum_i (y_i - \mu_y)^2}}$	Correlation coefficient, for $\mu_x = \mu_y = 0$ equivalent to the cosine distance [409].
Q13	$\frac{\sum_i x_i y_i - Km - m\sum_i x_i + m\sum_i y_i}{\sqrt{\left(\sum_i x_i^2 - Km^2 - 2m\sum_i x_i\right)\left(\sum_i y_i^2 + Km^2 - 2m\sum_i y_i\right)}}$	$m = \frac{x_{max} - x_{min}}{2}$, Cohen 1969 [58]
Q14	$\sum_{i=0}^{K-2} \left((x_i - x_{i+1}) - (y_i - y_{i+1})\right)^2$	Meehl index [263]
Q15	$\sum_i \min(x_i, y_i)$	Histogram intersection [361]
Q16	$\sum_i x_i \log\left(\frac{x_i}{y_i}\right)$	Kullback Leibler divergence [218]
Q17	$\sum_i (x_i - y_i) \log\left(\frac{x_i}{y_i}\right)$	Jeffrey divergence [175]
Q18	$\sum_i x_i \log\left(\frac{x_i}{y_i}\right)^2$	Exponential divergence [16]
Q19	$\frac{1}{2}\sum_i \frac{(x_i - y_i)^2}{x_i}$	Kagan divergence [179]

B.2 Predicate-Based Similarity Measures

Below, $a = \sum_i x \cap y, b = \sum_i x \setminus y, c = \sum_i y \setminus x, d = K - a - b - c, K = size(x) = size(y)$ of two predicate vectors $x, y \in F$ with $F = [s_l | s \in S \wedge l \in L]$ and $s = \{0, 1\}$.

Table B.2: Catalogue of Predicate-Based Similarity Measures.

No.	*Measure*	*Description*
P1	a	Number of co-occurences
P2	$\frac{a}{a+b+c+d}$	Russel and Rao 1940 [317]

...continued on next page

Table B.2: Catalogue of Predicate-Based Similarity Measures.

No.	*Measure*	*Description*
P3	$b + c$	Hamming distance [151]. The squared root of this measure is frequently called binary Euclidean distance.
P4	$\frac{a+d}{a+b+c+d}$	Simple matching coefficient [347] and [130]
P5	$a + d$	Complement of Hamming distance [130]
P6	$a - b - c$	Feature contrast model, Tversky 1977 [377]
P7	$\frac{a}{b+c}$	Kulczynski 1927 [217]
P8	$\frac{bc}{K^2}$	Pattern difference measure [7]
P9	$a - bc$	Browsing pattern [84]
P10	$a + \frac{b+c}{2}$	Köppen 1884 [207]
P11	$\frac{a}{a+b+c}$	Jaccard 1908 [170]
P12	$\frac{a}{a+max(b,c)}$	Braun 1932 [41]
P13	$\frac{a}{a+min(b,c)}$	Ecological coexistence coefficient, Simpson 1943 [340]
P14	$\frac{b+c}{4K}$	Variance dissimilarity measure [351]
P15	$\frac{(b-c)^2}{K^2}$	Baulieu 1989 [21], also size difference [351]
P16	$\frac{a+d}{b+c}$	Kulczynski 1927 [217]
P17	$\frac{Ka}{(a+b)(a+c)}$	Forbes 1925 [106]. Gilbert and Wells use the logarithm of this measure [121].
P18	$\frac{a-b-c+d}{K}$	Hamann 1961 [150]
P19	$\frac{K(b+c)-(b-c)^2}{K^2}$	Baulieu 1989 [21]
P20	$\frac{K(b+c)-(b-c)^2}{K^2}$	Binary shape difference [351]
P21	$\frac{a+d}{K+b+c}$	Rogers and Tanimoto 1960 [310]
P22	$\frac{a+d}{a+2b+2c+d}$	Gower and Legendre 1986 [135]
P23	$\frac{a-(a+b)(a+c)}{2}$	Steffensen 1934 [353]
P24	$\frac{2a}{2a+b+c}$	Czekanowski 1913 [64]. Sometimes divided by 2 [135].
P25	$\frac{4a}{4a+b+c}$	Sørensen 1948 [349]
P26	$\frac{a}{a+2b+2c}$	Sneath and Sokal 1963 [344]
P27	$\frac{8a}{8a+b+c}$	Anderberg 1973 [7]
P28	$\frac{a^2}{(a+b)(a+c)}$	Sorgenfrei 1958 [350]
P29	$\frac{a}{\sqrt{(a+b)(a+c)}}$	Independently defined by Driver and Kroeber [79] and Ochiai [282]
P30	$\frac{1}{2}(\frac{a}{a+b} + \frac{a}{a+c})$	Kulczynski 1927 [217]. Sometimes multiplied by 2 [163]
P31	ad	Retrieval pattern [84]
P32	$\frac{2(a+d)}{2(a+d)+b+c}$	Sneath and Sokal 1963 [344]. Sometimes divided by 2 [135].
P33	$\frac{a}{\frac{ab+ac}{2}+bc}$	Mountford 1962 [273]
P34	$\frac{a^2-bc}{(a+b)(a+c)}$	McConnaughey 1964 [258]
P35	$\frac{K(a-\frac{1}{2})^2}{(a+b)(a+c)}$	Jones and Curtis 1967 [177]
P36	$\frac{ad-bc}{K}$	Maron and Kuhns 1960 [247]
P37	$\frac{bc}{ad}$	Q_0 from Batagelj and Bren [20]. Inversely applied as odds ratio on [212]
P38	$\frac{Ka-(a+b)(a+c)}{Ka+(a+b)(a+c)}$	Tarwid 1960 [366]
P39	$\frac{a}{\sqrt{(a+b)(a+c)}} - \frac{a+max(b,c)}{2}$	Fager and McGowan 1963 [93]
P40	$2(ad - bc)$	Stuart's τ_c [358]
P41	$\frac{(a+b)(1-a-b)-c}{(a+b)(1-a-b)}$	Köppen 1870 [206]

...continued on next page

Table B.2: Catalogue of Predicate-Based Similarity Measures.

No.	*Measure*	*Description*
P42	$\frac{1}{2}(\frac{a}{a+b+c}+\frac{d}{b+c+d})$	Hawkins and Dotson 1975 [156]
P43	$\frac{a-(a+b)(a+c)}{a+min(b,c)-(a+b)(a+c)}$	Benini 1901 [28]
P44	$\frac{a-(a+b)(a+c)}{a+b+c-(a+b)(a+c)}$	Gilbert's coefficient [120]
P45	$\frac{\sqrt{ad}+a-b-c}{\sqrt{ad}+a+b+c}$	Baroni-Urbani and Buser [17]
P46	$\frac{a-(a+b)(a+c)}{1-\frlvert b-c\frrvert -(a+b)(a+c)}$	Modified Gini index [124]
P47	$\frac{ad-bc}{(a+c)(b+d)}$	Peirce 1884 [292]
P48	$\frac{2(ad-bc)}{K(2a+b+c)}$	Coefficient of arithmetic means, Kuhns 1965 [216]
P49	$\frac{a-(a+b)(a+c)}{\sqrt{(1-(a+b)^2)(1-(a+c)^2)}}$	Gini index [123]
P50	$\frac{a-(a+b)(a+c)}{(a+b)(c+d)(a+c)(b+d)}$	Eyraud 1936 [92]
P51	$\frac{2min(a,d)-b-c}{2min(a,d)+b+c}$	Goodman and Kruskal 1954 [131]
P52	$\frac{ab+bc}{ab+2bc+cd}$	Peirce 1884 [292]
P53	$\cos\frac{180\sqrt{bc}}{\sqrt{ad}+\sqrt{bc}}$	Pearson and Heron 1913 [291]
P54	$\frac{max(a,b)+max(c,d)-max(a+c,b+d)}{1-max(a+c,b+d)}$	Relative decrease of error probability [145]
P55	$\frac{max(a,c)+max(b,d)-max(a+b,c+d)}{1-max(a+b,c+d)}$	Goodman and Kruskal 1954 [131] in analogy to [145]
P56	$\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$	Michael 1920 [265]
P57	$\frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	Sneath and Sokal 1963 [344]
P58	$\frac{ad-bc}{ad+bc}$	Yule's Q 1911 [409]
P59	$\frac{ad-bc}{min((a+b)(a+c),(b+d)(c+d))}$	Cole 1949 [59]
P60	$\frac{ad-bc}{min((a+b)(b+d),(a+c)(c+d))}$	Loevinger's H [233]
P61	$\frac{a+d-max(a,d)-\frac{b+c}{2}}{1-max(a,d)-\frac{b+c}{2}}$	Goodman and Kruskal 1954 [131]
P62	$\frac{2(ad-bc)}{(a+b)(c+d)+(a+c)(b+d)}$	Maxwell and Pilliner 1968 [253]
P63	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	Pearson 1926 [290]. Goodman and Kruskal [132] suggest the squared Pearson coefficient.
P64	$\frac{\sqrt{ad}+a}{\sqrt{ad}+a+b+c}$	Baroni-Urbani and Buser [17]
P65	$\frac{1}{4}\cdot(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{b+d}+\frac{d}{c+d})$	Sneath and Sokal 1963 [344]
P66	$\frac{4(ad-bc)-(b-c)^2}{(2a+b+c)(b+c+2d)}$	Scott 1955 [328]
P67	$\frac{\sqrt{ad}-\sqrt{bc}}{\sqrt{ad}+\sqrt{bc}}$	Yule's Y coefficient [410]
P68	$\frac{(ad)^{\frac{3}{4}}-(bc)^{\frac{3}{4}}}{(ad)^{\frac{3}{4}}+(bc)^{\frac{3}{4}}}$	Digby 1983 [76]
P69	$\frac{ad-bc}{K(1-\frac{a}{(a+b)(a+c)})(2a+b+c-\frac{(a+b)*(a+c)}{K})}$	Proportion of overlap above independence, Kuhns 1965 [216]
P70	$\frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)}$	Cohen's κ [57]
P71	$\frac{(ad-bc)((a+b)(b+d)+(a+c)(c+d))}{2(a+b)(a+c)(b+d)(c+d)}$	Fleiss 1975 [101]
P72	$\sqrt{\frac{2(ad-bc)}{(ad-bc)^2-(a+b)(c+d)(a+c)(b+d)}}$	Mean square contingency. Cole 1949 [59]
P73	$K(\frac{a^2}{(a+b)(a+c)}+\frac{b^2}{(a+b)(b+d)}+\frac{c^2}{(a+b)(c+d)}+\frac{d^2}{(b+d)(c+d)}-1)$	Chi-square statistics [411], sometimes also divided by K [60] and/or square rooted [188]
P74	$\frac{max(a,b)+max(c,d)+max(a,c)+max(b,d)}{2-max(a+c,b+d)-max(a+b,c+d)}-\frac{max(a+c,b+d)+max(a+b,c+d)}{2-max(a+c,b+d)-max(a+b,c+d)}$	Goodman and Kruskal 1954 [131] in analogy to [145], sometimes also divided by K^2. Redundantly written for better readability.

...continued on next page

Table B.2: Catalogue of Predicate-Based Similarity Measures.

No.	*Measure*	*Description*
P75	$\frac{\frac{(a-(a+b)(a+c))^2+(b-(a+b)(b+d))^2}{a+b}}{1-(a+c)^2-(b+d)^2}+$ $\frac{\frac{(c-(a+c)(c+d))^2+(d-(b+d)(c+d))^2}{c+d}}{1-(a+c)^2-(b+d)^2}$	Goodman's and Kruskal's τ_b Goodman 1954 [131]. Redundantly written for better readability.

B.3 Similarity Meta-Models

No.	*Measure*	*Description*
M1	$m_i(x,y)+\delta_x+\delta_y$	Density model, Krumhansl 1978 [214]. See Chapter 17 for a discussion of the density terms δ_x, δ_y.
M2	$\frac{\sqrt{2K}-m_i(x,y)}{\sqrt{2K}+m_i(x,y)}$	Catell 1949 [50]
M3	$log\left(\frac{m_i(x,y)d_i(a,b)}{m_i(a,y)d_i(x,b)}\right)$	Cayley Klein model [196], where a, b are intersection points of the geodesic line through x, y with the fundamental conic section.
M4	$m_i(x,y)^{K-a}$	Product rule defined by Estes [90], where a is the number of communalities of x, y, without weights.
M5	$\inf_x\left(\sup_y\left(m_i(x,y)\right)\right)$	Bottleneck distance.
M6	$\max\left(\sup_x\left(\inf_y\left(m_i(x,y)\right)\right), \sup_y\left(\inf_x\left(m_i(x,y)\right)\right)\right)$	Hausdorff model (e.g. [96])
M7	$\inf_{\bar{x}\in perm(x)}\left(\frac{\sum_j m_i(\bar{x}_j, y_j)c(\bar{x}_j, y_j)}{K}\right)$	Mallows distance [240] (a.k.a. Wasserstein distance, earth mover's distance), where $c()$ is the cost function weighting the distances.
M8	$\inf_{x,y}\max_t\left(m_i(x(t), y(t))\right)$	Fréchet distance for two reparameterizations $x, y \in [0,1]$ of t (e.g. curves in parameterized form).

Table B.3: Catalogue of Similarity Meta-Models.

B.4 Dual Process Models

Operation	*Predicate*	*Quantity*	*FFCM*	*QM*
$x \cap y$	a	$\sum_i x_i y_i$	$min(x, y)$	$\frac{x+y}{2} > q \rightarrow \frac{x+y}{2} : 0$
$x \setminus y$	b	$\sum_i (x_i - y_i) \forall x_i > y_i$	$max(x - y, 0)$	$x - y > q \rightarrow x - y : 0$

Table B.4: Catalogue of Operators of Dual Process Models.

Appendix C

Media Programming Tools

In the tables of this chapter we employ the symbol ⊕ for a fully present feature, ⊖ for an absent feature and ⊘ for a feature that is partially present or under development.

C.1 General Properties

Property	*Matlab*	*OpenCV*	*R*	*Weka*
Audio import and export	⊕	⊖	⊖	⊖
Basic audio processing	⊕	⊖	⊖	⊖
Basic image processing	⊕	⊕	⊖	⊖
Basic text processing	⊘	⊖	⊖	⊖
Basic video processing	⊕	⊕	⊖	⊖
Biosignal import/export	⊘	⊖	⊖	⊖
Command history	⊕	⊖	⊕	⊖
Graphical user interface	⊕	⊖	⊘	⊕
Image import/export	⊕	⊕	⊖	⊖
License	commercial	free	free	free
Programming language	proprietary	C	proprietary	Java
Symbolic import/export	⊘	⊖	⊕	⊖
User interface designer	⊕	⊘	⊖	⊖
Variable editor	⊕	⊖	⊕	⊘
Video import/export	⊕	⊕	⊖	⊖

Table C.1: General Properties of Media Understanding Software.

C.2 Feature Transformations

Weka is not considered in this table, because this software does not provide any functions for feature transformation.

Property	*Matlab*	*OpenCV*	*R*
Biosignal features	⊕	⊖	⊖
Color features	⊕	⊕	⊖
General signal processing	⊕	⊘	⊘
Integral transforms	⊕	⊕	⊕
Local features	⊕	⊕	⊖
Motion features	⊕	⊕	⊖
MPEG-7 descriptions	⊘	⊘	⊖
Psychophysical transforms	⊘	⊖	⊖
Shape features	⊘	⊕	⊖
Spectral audio features	⊕	⊕	⊖
Template matching	⊘	⊕	⊖
Texture features	⊕	⊕	⊖
Time-based audio features	⊕	⊖	⊖

Table C.2: Feature Transformations in Matlab, OpenCV and R.

C.3 Information Filtering and Visualization

Property	*Matlab*	*OpenCV*	*R*	*Weka*
Correlation analysis	⊕	⊕	⊕	⊖
Factor analysis	⊕	⊕	⊕	⊖
Kalman filter	⊕	⊕	⊘	⊖
Normalization	⊕	⊕	⊕	⊕
Regression	⊕	⊕	⊕	⊖
Source separation	⊕	⊖	⊘	⊖
Statistical moments	⊕	⊕	⊕	⊕
Statistical testing	⊕	⊖	⊕	⊖
Variable plotter	⊕	⊘	⊕	⊕

Table C.3: Information Filtering and Visualization in Media Understanding Software.

C.4 Categorization and Evaluation

Property	*Matlab*	*OpenCV*	*R*	*Weka*
Bayesian classifier	⊕	⊕	⊕	⊕
Bayesian network	⊘	⊖	⊕	⊕
Boosting	⊕	⊕	⊕	⊘
Cluster analysis	⊕	⊖	⊕	⊕
Decision tree	⊘	⊕	⊕	⊖
Dynamic time warping	⊕	⊖	⊖	⊖
Expectation maximization	⊕	⊕	⊕	⊖
Gaussian mixture model	⊕	⊖	⊕	⊕
Gibbs sampling	⊘	⊖	⊘	⊖
Hidden Markov model	⊕	⊖	⊕	⊕
K-means	⊕	⊘	⊕	⊕
K-nearest neighbor	⊕	⊕	⊕	⊕
Linear discriminant analysis	⊕	⊖	⊕	⊕
Perceptron network	⊘	⊕	⊕	⊕
Radial basis function	⊘	⊖	⊕	⊕
Self-organizing map	⊕	⊖	⊕	⊘
Support vector machine	⊕	⊕	⊘	⊕
Vector space model	⊕	⊖	⊕	⊘

Table C.4: Categorization in Media Understanding Software.

C.5 Mobile Implementation

Property	*Android*	*iPhone*	*Symbian*
Audio capturing	⊕	⊕	⊕
Localization system	⊕	⊕	⊕
Socket communication	⊕	⊕	⊕
Video capturing	⊕	⊕	⊕

Table C.5: Properties of Environments for Mobile Implementation of Media Understanding Applications.

Bibliography

[1] BR Abidi, NR Aragam, Y Yao, and MA Abidi. Survey and analysis of multimodal sensor planning and integration for wide area surveillance. *ACM Comput. Surv.*, 41:7:1–7:36, 2009.

[2] D Adams. *The Hitchhiker's Guide to the Galaxy.* Pan, 1979.

[3] PS Addison. *The Illustrated Wavelet Transform Handbook.* Taylor and Francis, 2002.

[4] FX Albizuri, A Danjou, M Grana, J Torrealdea, and MC Hernandez. The high-order boltzmann machine: Learned distribution and topology. *IEEE Transactions on Neural Networks*, 6(3):767–770, 1995.

[5] T Alieva and MJ Bastiaans. Wigner distribution and fractional fourier transform. *Proceedings Symposium on Signal Processing and its Applications*, 1:168–169, 2001.

[6] E Alpaydin. *Introduction to Machine Learning.* MIT Press, 2004.

[7] M Anderberg. *Cluster Analysis for Applications.* Academic Press, New York, 1973.

[8] NH Anderson and DM Titterington. Beyond the binary boltzmann machine. *IEEE Transactions on Neural Networks*, 6(5):1229–1236, 1995.

[9] PMG Apers, HM Blanken, and MAW Houtsma. *Multimedia Databases in Perspectives.* Springer, 1997.

[10] FG Ashby and NA Perrin. Toward a unified theory of similarity and recognition. *Psychological Review*, 95(1):124–150, 1988.

[11] F Attneave. Dimensions of similarity. *American Journal of Psychology*, 62:516–556, 1950.

[12] F Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.

[13] O Avaro and P Salembier. Mpeg-7 systems: Overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):760–764, 2001.

[14] R Baeza-Yates and B Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 2011.

[15] N Barley. *The Innocent Anthropologist : Notes from a Mud Hut*. Waveland, 2000.

[16] OE Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. Wiley, 1978.

[17] C Baroni-Urbani and MW Buser. Similarity of binary data. *Systematic Zoology*, 25:251–259, 1976.

[18] R Barthes. *Image-Music-Text*. Hill and Wang, 1978.

[19] S Basu, I Davidson, and K Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman and Hall, 2008.

[20] V Batagelj and M Bren. Comparing resemblance measures. *Journal of Classification*, 12:73–90, 1995.

[21] FB Baulieu. A classification of presence/absence based dissimilarity coefficients. *Journal of Classification*, 6:233–246, 1989.

[22] A Beach. *Real World Video Compression*. Peachpit Press, 2008.

[23] SS Beauchemin and JL Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.

[24] A Bejan and S Lorente. The constructal law origin of the logistics s curve. *Journal of Applied Physics*, 110(2):24901–24904, 2011.

[25] R Bellman and S Dreyfus. *Dynamic Programming*. Princeton University Press, 2010.

[26] S Belongie, J Malik, and J Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[27] N Ben Mustapha, HB Zghal, MA Aufaure, and H ben Ghezala. Semantic search using modular ontology learning and case-based reasoning. In *Proceedings of the 2010 EDBT/ICDT Workshops*, pages 3:1–3:12, New York, NY, USA, 2010. ACM.

[28] R Benini. *Principles of Demography (in Italian)*. Manuali Barbara di Scienze Giuridiche Sociali e Politiche, 1901.

[29] W Benjamin. *The Work of Art in the Age of Its Technological Reproducibility, and Other Writings on Media*. Harvard University Press, 2008.

[30] D Bernstein. *The Design and Implementation of Multimedia Software With Examples in Java*. Jones and Bartlett, 2010.

[31] MW Berry and M Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM, 2005.

[32] A Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.

[33] A Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, 1999.

[34] A Del Bimbo and P Pala. Content-based retrieval of 3d models. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2:20–43, 2006.

[35] CM Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[36] A Blake and M Isard. *Active Contours*. Springer, 1998.

[37] M Bober. Mpeg-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, 2001.

[38] H Böll. *The Collected Stories of Heinrich Böll*. Melville House, 2011.

[39] S Boriah, V Chandola, and V Kumar. Similarity measures for categorical data: A comparative evaluation. *Proceedings of SIAM Data Mining Conference*, pages 1–12, 2008.

[40] GEP Box, GM Jenkins, and GC Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, 1994.

[41] J Braun-Blanquet. *Plant Sociology: The Study of Plant Communities*. McGraw-Hill, 1932.

[42] H Breu, J Gil, D Kirkpatrick, and M Werman. Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, 1995.

[43] V Britanak, PC Yip, and KR Rao. *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*. Academic Press, 2006.

[44] JL Bruning and BL Kintz. *Computational Handbook of Statistics*. Allyn and Bacon, 1997.

[45] P Burrascano. Learning vector quantization for the probabilistic neural network. *IEEE Transactions on Neural Networks*, 2(4):458–461, 1991.

[46] EJ Candes and DL Donoho. Ridgelets: A key to higher-dimensional intermittency? *Philosophical Transactions of the Royal Society*, pages 2495–2509, 1999.

[47] Pedro Cano, Markus Koppenberger, and Nicolas Wack. Content-based music audio recommendation. pages 211–212, New York, NY, USA, 2005.

[48] R Carter, S Aldridge, M Page, S Parker, and C Frith. *The Human Brain Book*. DK ADULT, 2009.

[49] G Casella and RL Berger. *Statistical Inference*. Duxbury Press, 2001.

[50] RB Catell. r_p and other coefficients of pattern similarity. *Psychometrica*, 14:279–298, 1949.

[51] TF Chan and LA Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[52] G Chandler. *Cut by Cut: Editing Your Film or Video*. Michael Wiese Productions, 2006.

[53] SC Chen, RL Kashyap, and A Ghafoor. *Semantic Models for Multimedia Database Searching and Browsing*. Kluwer, 2000.

[54] V Cherkassky and FM Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley, 2007.

[55] PS Clark. An extension of the coefficient of divergence for use with multiple characters. *Copeia*, 2:61–64, 1952.

[56] H Cohen and C Lefebvre. *Handbook of Categorization in Cognitive Science*. Elsevier, 2005.

[57] J Cohen. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220, 1968.

[58] J Cohen. A profile similarity coefficient invariant over variable reflection. *Psychological Bulletin*, 71:281–284, 1969.

[59] LC Cole. The measurement of interspecific association. *Ecology*, 30:411–424, 1949.

[60] H Cramér. *Mathematical Models of Statistics*. Princton University Press, 1946.

[61] N Cristianini and J Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[62] N Cristianini and JS Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[63] Bin Cui, H. V. Jagadish, Beng Chin Ooi, and Kian-Lee Tan. Compacting music signatures for efficient music retrieval. In *Proceedings of the 11th International Conference on Extending Database Technology*, pages 229–240, New York, NY, USA, 2008.

[64] J Czekanowski. Overview over the application of statistical methods for anthropology (in polish). *Prace Towarzystwa Naukowego Warszawskiego*, 5, 1913.

[65] A da Silva Meyer, AA Franco Garica, A Pereira de Souza, and C Lopez de Souza. Comparison of similarity coefficients used for cluster analysis with dominant markers in maize. *Genetics and Molecular Biology*, 27(1):83–91, 2004.

[66] SB Dalirsefat, A da Silva Meyer, and SZ Mirhoseini. Comparision of similarity coefficients used for cluster analysis with amplified fragment length polymorphism markers in the silkworm. *Journal of Insect Science*, 9:1–8, 2009.

[67] R Datta, D Joshi, J Li, and JZ Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[68] ER Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann, 2005.

[69] TA Davis. *MATLAB Primer*. CRC Press, 2010.

[70] A de la Torre, AM Peinado, JC Segura, JL Perez-Cordoba, MC Benitez, and AJ Rubio. Histogram equalization of speech representation for robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(3):355–366, 2005.

[71] SR Deans. *The Radon Transform and Some of Its Applications*. Dover Publications, 2007.

[72] L Debnath and D Bhatta. *Integral Transforms and Their Applications*. Chapman and Hall, 2006.

[73] E Dedrick and D Lau. A kalman-filtering approach to high dynamic range imaging for measurement applications. *IEEE Transactions on Image Processing*, 21(2):527–536, 2012.

[74] PA Devijver and J Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.

[75] LR Dice. Measures of the amount of ecological association between species. *Ecology*, 26:297–302, 1945.

[76] PGN Digby. Approximating the tetrachoric correlation coefficient. *Biometrics*, 39:753–757, 1983.

[77] A Diplaros, T Gevers, and I Patras. Combining color and shape information for illumination-viewpoint invariant object recognition. *IEEE Transactions on Image Processing*, 15(1):1–11, 2006.

[78] S Dominich. *The Modern Algebra of Information Retrieval*. Springer, 2010.

[79] HE Driver and AL Kroeber. Quantitative expression of cultural relationships. *University of California Publications in American Archaeology and Ethnology*, 31:211–256, 1932.

[80] ST Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230, 2004.

[81] U Eco. *A Theory of Semiotics*. Indiana University Press, 1978.

[82] JP Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

[83] H Eidenberger. How good are the visual mpeg-7 features? In *Proceedings of IEEE Visual Communications and Image Processing Conference*, Lugano, Switzerland, 2003. SPIE.

[84] H Eidenberger. Evaluation and analysis of similarity measures for content-based visual information retrieval. *ACM Multimedia Systems*, 12(2):71–87, 2006.

[85] H Eidenberger. Descriptor evaluation for visual information retrieval using self-organising maps and other statistical methods. *Multimedia Tools and Applications*, 35(3):241–258, 2007.

[86] H Eidenberger. *Fundamental Media Understanding*. atpress, 2011.

[87] P Ekman. *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life*. Holt Paperbacks, 2007.

[88] R ElAttar. *Lecture Notes on Z-Transform*. Lulu.com, 2006.

[89] E England and A Finney. *Managing Multimedia*. Addison Wesley, 2002.

[90] WK Estes. *Classification and Cognition*. Oxford University Press, New York, 1994.

[91] BS Everitt, S Landau, M Leese, and D Stahl. *Cluster Analysis*. Wiley, 2011.

[92] H Eyraud. The principles of the measurement of correlation (in french). *Annales de l'Universite de Lyon, Serie III, Section A*, 1:30–45, 1936.

[93] EW Fager and JA McGowan. Zooplankton species groups in the north pacific. *Science*, 140:453–460, 1963.

[94] K Falconer. *Fractal Geometry*. Wiley, 2006.

[95] H Fastl and E Zwicker. *Psychoacoustics: Facts and Models*. Springer, 2006.

[96] H Federer. *Geometric Measure Theory*. Springer-Verlag, Berlin, 1969.

[97] D Feng, WC Siu, and HJ Zhang. *Multimedia Information Retrieval and Management*. Springer, 2010.

[98] S Few. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, 2009.

[99] SE Fienberg. *The Analysis of Cross-Classified Categorical Data*. Springer, 2007.

[100] GA Fink. *Markov Models for Pattern Recognition: From Theory to Applications*. Springer, 2007.

[101] JL Fleiss. Measuring agreement between two judges on the presence or absence of a trait. *Biometrics*, 31:651–659, 1975.

[102] R Fletcher. *Practical Methods of Optimization*. Wiley, 2000.

[103] J Flusser, B Zitova, and T Suk. *Moments and Moment Invariants in Pattern Recognition*. Wiley, 2009.

[104] V Flusser. *The Shape of Things*. Reaktion Books, 1999.

[105] J Foote. *An Overview of Audio Information Retrieval.* Springer, 1999.

[106] SA Forbes. Method of determining and measuring the associative relations of species. *Science*, 61:524, 1925.

[107] RA Frost. Realization of natural language interfaces using lazy functional programming. *ACM Comput. Surv.*, 38, 2006.

[108] D Fudenberg and J Tirole. *Game Theory.* MIT Press, 1991.

[109] N Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[110] B Furht, SW Smoliar, and HJ Zhang. *Video and image processing in multimedia systems.* Springer, 1995.

[111] MA Gandhi and L Mili. Robust kalman filter based on a generalized maximum-likelihood-type estimator. *IEEE Transactions on Signal Processing*, 58(5):2509–2520, 2010.

[112] WR Garner. *The Processing of Information and Structure.* Wiley, 1974.

[113] KR Gegenfurtner. *Brain and Perception (in German).* Fischer, 2006.

[114] A Gelman. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge University Press, 2006.

[115] LQ Geng and HJ Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3):1–32, 2006.

[116] RJ Gennaro. *The Consciousness Paradox: Consciousness, Concepts, and Higher-Order Thoughts.* MIT Press, 2011.

[117] D Gentner and A Markman. Structure mapping in analogy and similarity. *American Psychologist*, 52(1):45–56, 1997.

[118] W Gerstner and WM Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity.* Cambridge University Press, 2002.

[119] A Ghosh and N Petkov. Robustness of shape descriptors to incomplete contour representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1793–1804, 2005.

[120] GK Gilbert. Finley's tornado predictions. *American Meteorological Journal*, 1:166–172, 1884.

[121] N Gilbert and TCE Wells. Analysis of quadrat data. *Journal of Ecology*, 54:675–685, 1966.

[122] R Gilmore. *Lie Groups, Lie Algebras, and Some of Their Applications*. Dover, 2006.

[123] C Gini. Index of homophily and likeness and its relation to the correlation coefficient and the indices of attraction (in italian). *Atti del Reale Instituto Veneto di Scienze*, 74(2):583–610, 1915.

[124] C Gini. New contribution to the theory of statistical relationships (in italian). *Atti del Reale Instituto Veneto di Scienze*, 74(2):1903–1942, 1915.

[125] DE Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[126] Erich Goldmeier. Similarity in visually perceived forms. *Psychological Issues*, 8(1):135, 1972.

[127] EB Goldstein. *Sensation and Perception*. Wadsworth Publishing, 2009.

[128] R Goldstone. *Similarity*. Indiana University Technical Report, 2005.

[129] RC Gonzalez and RE Woods. *Digital Image Processing*. Prentice Hall, 2007.

[130] DW Goodall. The distribution of the matching coefficient. *Biometrics*, 23:647–656, 1967.

[131] LA Goodman and WH Kruskal. Measures of association for cross classifications. *American Statistical Association Journal*, 49:732–764, 1954.

[132] LA Goodman and WH Kruskal. Measures of association for cross classifications. ii: Further discussion and references. *American Statistical Association Journal*, 54:123–163, 1959.

[133] N Goodman. *Problems and Projects*. Hackett Publishing Company, Indianapolis, 1972.

[134] RL Gorsuch. *Factor Analysis*. Psychology Press, 1983.

[135] JC Gower and P Legendre. Metric and euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3:5–48, 1986.

[136] JG Gower. Multivariate analysis and multidimensional geometry. *The Statistician*, 17:13–25, 1967.

[137] M Graczyk, T Lasota, and B Trawinski. Comparative analysis of premises valuation models using keel, rapidminer, and weka. *Lecture Notes in Computer Science*, 5796:800–812, 2009.

[138] M Granitzer, M Lux, and M Spaniol. *Multimedia Semantics – The Role of Metadata*. Springer, 2008.

[139] RM Gray. *Entropy and Information Theory*. Springer, 2011.

[140] JM Grey and JW Gordon. Perceptual effects of spectral modifications on musical timbres. *Journal of the Acoustic Society of America*, 63(5):1493–1500, 1978.

[141] AM Grigoryan and SS Agaian. *Multidimensional Discrete Unitary Transforms*. CRC Press, 2003.

[142] GA Gscheider. *Psychophysics*. Lawrence Erlbaum, 1997.

[143] Y Guan, X Wang, and Q Wang. A new measurement of systematic similarity. *IEEE Transactions on Systems, Man, and Cybernetics*, 38(4):743–758, 2008.

[144] S Gusenbauer. *Relevance Feedback in Information Retrieval*. VDM, 2004.

[145] L Guttman. An outline of the statistical theory of prediction. In P Horst, editor, *The Prediction of Personal Adjustment*, pages 253–318, New York, 1941. Social Science Research Council Bulletin 48.

[146] I Guyon and A Elisseef. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[147] U Hahn and TM Bailey. What makes words sound similar? *Cognition*, 97:227–267, 2005.

[148] U Hahn, N Chater, and L Richardson. Similarity as transformation. *Cognition*, 87:1–32, 2003.

[149] U Hahn and M Ramscar. *Similarity and Categorization*. Oxford University Press, 2001.

[150] U Hamann. Feature inventory and relationships of farinosae (in german). *Willdenowia Journal*, 2:639–768, 1961.

[151] RW Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2):147–160, 1950.

[152] A Hampapur, R Jain, and T Weymouth. Digital video segmentation. *ACM Multimedia Conference Proceedings*, pages 357–364, 1994.

[153] JW Han, M Kamber, and J Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[154] D Harman and G Marchionini. *Information Retrieval Evaluation*. Morgan and Claypool, 2011.

[155] MH Hassoun. *Fundamentals of Artificial Neural Networks*. Bradford Books, 2003.

[156] RP Hawkins and VA Dotson. Reliability scores that delude: An alice in wonderland trip through misleading characteristics of interobserver agreement scores in interval recording. In Ramp E and Semb G, editors, *Behavior Analysis: Areas of Research and Applications*, pages 359–376, Englewood Cliffs, 1975. Prentice-Hall.

[157] S Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 2008.

[158] R Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, 2001.

[159] TC Hodgman, A French, and DR Westhead. *Bioinformatics*. Taylor and Francis, 2010.

[160] RV Hogg and E Tanis. *Probability and Statistical Inference*. Prentice Hall, 2009.

[161] D Howard and J Angus. *Acoustics and Psychoacoustics*. Focal Press, 2009.

[162] Jia-Lien Hsu, Arbee L. P. Chen, Hung-Chen Chen, and Ning-Han Liu. The effectiveness study of various music information retrieval approaches. In *Proceedings of the eleventh International Conference on Information and Knowledge Management*, pages 422–429, New York, NY, USA, 2002.

[163] Z Hubálek. Coefficients of association and similarity, based on binary (presence-absence) data: An evaluation. *Biological Reviews*, 57:669–689, 1981.

[164] DH Hubel and TN Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[165] DA Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 1952:1098–1102, 40.

[166] D Hume. *An Enquiry Concerning Human Understanding*. CreateSpace, 2011.

[167] A Hyvärinen, J Karhunen, and E Oja. *Independent component analysis*. Wiley, 2001.

[168] S Imai. Pattern similarity and cognitive transformations. *Acta Psychologica*, 41:433–447, 1977.

[169] EM Izhikevich. *Dynamical Systems in Neuroscience*. MIT Press, 2010.

[170] P Jaccard. New research on plant distribution (in french). *Bulletin de la Société Vaudoise Sciences Naturelles*, 44:223–270, 1908.

[171] F Jäkel, B Schölkopf, and FA Wichmann. Generalization and similarity in exemplar models of categorization: Insights from machine learning. *Psychonomic Bulletin and Review*, 15(2):256–271, 2008.

[172] PK Janert. *Data Analysis with Open Source Tools*. O'Reilly Media, 2010.

[173] S Janson and J Vegelius. Measures of ecological association. *Oecologia*, 49:371–376, 1981.

[174] S Jeannin and A Divakaran. Mpeg-7 visual motion descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):720–724, 2001.

[175] H Jeffreys. *Theory of Probability*. Oxford University Press, 1948.

[176] NF Johnson. *Simply Complexity: A Clear Guide to Complexity Theory*. Oneworld, 2009.

[177] PF Jones and RM Curtis. A framework for comparing term association measures. *American Documentation*, 18:153–161, 1967.

[178] D Jurafsky. *Speech and Language Processing*. Prentice Hall, 2008.

[179] AM Kagan. On the theory of fisher amount of information. *Soviet Mathematics Doclady*, 4(4):991–993, 1963.

[180] M Kahn. *Technical Analysis Plain and Simple*. Pearson Education, 2010.

[181] D Kahneman, P Slovic, and A Tversky. *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press, New York, 1982.

[182] D Kahnemann and DT Miller. Norm theory: Comparing reality to its alternatives. *Psychological Review*, 93(2):136–153, 1986.

[183] D Kahnemann, P Slovic, and A Tversky. *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press, 1982.

[184] RE Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–45, 1960.

[185] B Kamich. *Chart Patterns*. Bloomberg Press, 2009.

[186] ER Kandel, JH Schwartz, and TM Jesell. *Principles of Neural Sciences*. McGraw Hill, 2000.

[187] MJ Kearns and U Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[188] MG Kendall. *The Advanced Theory of Statistics*. Charles Griffin and Co., 1948.

[189] P Kennedy. *A Guide to Econometrics*. Blackwell Publishing, 2009.

[190] S Khan. Lucas-kanade optical flow algorithm, http://www.cs.ucf.edu/ vision/code/optical_flow/lucas_kanade.zip, last visited October 31, 2013.

[191] HG Kim, N Moreau, and T Sikora. *MPEG-7 Audio and Beyond*. Wiley, 2005.

[192] T Kim, S Lee, and J Paik. Combined shape and feature-based video analysis and its application to non-rigid object tracking. *IEEE Transactions on Image Processing*, 5(1):87–100, 2011.

[193] FAA Kingdom and N Prins. *Psychophysics: A Practical Introduction*. Academic Press, 2009.

[194] S. Kiranyaz and M. Gabbouj. Generic content-based audio indexing and retrieval framework. *IEE Vision, Image and Signal Processing Proceedings*, 153(3):285 – 297, 2006.

[195] F Kittler and J Johnston. *Literature, Media, Information Systems*. Routledge, 1997.

[196] F Klein. On so-called non-euclidean geometry (in german). *Annals of Mathematics (Mathematische Annalen)*, 4:573–625, 1871.

[197] V Klema and A Laub. The singular value decomposition. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.

[198] H Kobayashi, Y Okouchi, and S Ota. Image retrieval system using kansei features. *Lecture Notes in Computer Science*, 1531:626–635, 1998.

[199] J Koehler, N Morgan, H Hermansky, HG Hirsch, and G Tong. Integrating rasta-plp into speech recognition. In *IEEE Acoustics, Speech, and Signal Processing Conference*, pages 421–424, Adelaide, 1994.

[200] T Kohonen. The neural phonetic typewriter. *IEEE Computer*, 21(3):11–22, 1988.

[201] T Kohonen. *Self-Organizing Maps*. Springer, 2000.

[202] PA Kolers. *Aspects of Motion Perception*. Pergamon Press, 1972.

[203] B Kollmeier, G Klump, V Hohmann, U Langemann, M Mauermann, S Uppenkamp, and J Verhey. *Hearing – From Sensory Processing to Perception*. Springer, 2007.

[204] J Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Francisco, 1993.

[205] S Komatineni, D MacLean, and S Hashimi. *Pro Android 3*. Apress, 2011.

[206] W Köppen. The sequence of unperiodic weather events by the principles of probability theory (in german). *Akademiia Nauk Repertorium f?r Meteorologie*, 2:189–238, 1870.

[207] W Köppen. A rational method for the evaluation of weather forecasts (in german). *Meteorologische Zeitschrift*, 1:397–404, 1884.

[208] I Koprinska and S Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16:477–500, 2001.

[209] T Koski. *Hidden Markov Models of Bioinformatics*. Springer, 2002.

[210] GJ Kowalski and MT Maybury. *Information Storage and Retrieval Systems*. Kluwer, 2000.

[211] HC Kraemer. Assessment of 2x2 associations: Generalization of signal-detection methodology. *American Statistician*, 42(1):37–49, 1988.

[212] HC Kraemer. Reconsidering the odds ratio as a measure of 2x2 association in a population. *Statistics in Medicine*, 23:257–270, 2004.

[213] HC Kraemer, VS Periyakoil, and A Noda. Kappa coefficients in medical research. *Tutorials in Biostatistics*, 1:85–105, 2004.

[214] C Krumhansl. Concerning the applicability of geometric models to similarity data. *Psychological Review*, 85(5):445–463, 1978.

[215] JB Kruskal. Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[216] JL Kuhns. The continuum of coefficients of association. In Stevens ME, Giuliano VE, and Heilprin LB, editors, *Statistical Association Methods for Mechanized Documentation*, pages 33–40, Washington, 1965. NBS Misc Publication.

[217] S Kulczynski. Supplement ii. *Bulletin International de l'Acadamie Polonaise des Sciences et des Lettres, Classe des Sciences Mathematiques et Naturelles, Série B (Sciences Naturelles)*, pages 57–203, 1927.

[218] S Kullback and RA Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[219] GN Lance and WT Williams. Mixed data classificatory programs. *Agglomerative Systems Australian Company Journal*, 9:373–380, 1967.

[220] WB Langdon and R Poli. *Foundations of Genetic Programming*. Springer, 2010.

[221] S Lankton and A Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008.

[222] VF Leavers. *Shape Detection in Computer Vision Using the Hough Transform*. Springer, 1992.

[223] EL Lehmann and JP Romano. *Testing Statistical Hypotheses*. Springer, 2010.

[224] GG Lendaris, K Mathia, and R Saeks. Linear hopfield networks and constrained optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(1):114–118, 1999.

[225] A Lesk. *Introduction to Bioinformatics*. Oxford University Press, 2008.

[226] MJ Lesot, M Rifqi, and H Benhadda. Similarity measures for binary and numerical data: A survey. *Knowledge Engineering and Soft Data Paradigms*, 1:63–84, 2009.

[227] MS Lew. *Principles of Visual Information Retrieval*. Springer, 2001.

[228] D Lin. An information-theoretic definition of similarity. *Proceedings of 15th International Machine Learning Conference*, 15:296–304, 1998.

[229] IJ Lin and SY Kung. *Video Object Extraction and Representation*. Kluwer, 2000.

[230] T Lindeberg. *Scale Space Theory in Computer Vision*. Springer, 2010.

[231] H Liu and H Motoda. *Computational methods of feature selection*. Chapman and Hall, 2008.

[232] TY Liu. *Learning to Rank for Information Retrieval*. Springer, 2011.

[233] JA Loevinger. The technique of homogeneous tests compared with some aspects of scale analysis and factor analysis. *Psychological Bulletin*, 45:507–530, 1948.

[234] B Logan. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval*, Plymouth, MA, 2000.

[235] D Lopresti. Optical character recognition errors and their effects on natural language processing. In *Proceedings of the second Workshop on Analytics for Noisy Unstructured Text Data*, pages 9–16, New York, NY, USA, 2008. ACM.

[236] EN Lorenz. *The Essence Of Chaos*. CRC Press, 1995.

[237] DG Lowe. Object recognition from local scale-invariant features. In *IEEE Computer Vision Conference*, pages 1150–1157. IEEE Press, 1999.

[238] RD Luce. *Individual Choice Behavior*. Dover Publications, Mineola NY, 1959.

[239] PC Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

[240] CL Mallows. A note on asymptotic joint normality. *Annals of Mathematical Statistics*, 43(2):508–515, 1972.

[241] B Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, 1982.

[242] BS Manjunath, JR Ohm, VV Vasudevan, and A Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[243] BS Manjunath, P Salembier, and T Sikora. *Introduction to MPEG-7: multimedia content description interface*. Wiley, 2002.

[244] CD Manning and H Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[245] E Margolis and S Laurence. *Concepts*. MIT Press, Cambridge MA, 1999.

[246] E Margolis and S Laurence. *Creations of the Mind: Theories of Artifacts and Their Representation*. 2007, Oxford University Press.

[247] ME Maron and JL Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7(3):216–244, 1960.

[248] D Marr, TA Poggio, and S Ullman. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Pres, 2010.

[249] M Marszaek and C Schmid. Spatial weighting for bag-of-features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2118–2125. IEEE Press, 2006.

[250] WL Martinez and AR Martinez. *Computational Statistics Handbook with MATLAB*. Chapman and Hall, 2007.

[251] A Martins. String kernels and similarity measures for information retrieval. *Priberam Technical Report*, 2006.

[252] MW Matlin. *Cognition*. Wiley, 2008.

[253] AE Maxwell and AEG Pilliner. Deriving coefficients of reliability and agreement for ratings. *British Journal of Mathematical and Statistical Psychology*, 21:105–116, 1968.

[254] MT Maybury. *Intelligent Multimedia Information Retrieval*. MIT Press, 1997.

[255] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Combination of audio and lyrics features for genre classification in digital audio collections. In *Proceeding of the 16th ACM International Conference on Multimedia*, pages 159–168, 2008.

[256] R Mazza. *Introduction to Information Visualization*. Springer, 2009.

[257] F Mcallen. *Charting and Technical Analysis*. CreateSpace, 2010.

[258] BH McConnaughey. The determination and analysis of plankton communities. *Marine Research of Indonesia Species*, 1:1–40, 1964.

[259] G McLachlan and D Peel. *Finite Mixture Models*. Wiley, 2000.

[260] M McLuhan. *Understanding Media: The Extensions of Man*. MIT Press, 1994.

[261] M McLuhan. *The Gutenberg Galaxy*. University of Toronto Press, 2011.

[262] CT Meadow, BR Boyce, and DH Kraft. *Text Information Retrieval Systems*. Emerald Group Publishing Limited, 2007.

[263] PE Meehl. The problem is epistemology, not statistics: Replace significance tests by confidence intervals and quantify accuracy of risky numerical predictions. In LL Harlow, SA Mulaik, and JH Steiger, editors, *What If There Were No Significance Tests?*, pages 393–425, Mahwah NJ, 1997. Erlbaum.

[264] W Metzger and L Spillmann. *Laws of Seeing*. MIT Press, 2009.

[265] EL Michael. Marine ecology and the coefficient of association: A plea in behalf of quantitative biology. *Journal of Animal Ecology*, 8:54–59, 1920.

[266] RF Mihalcea and DR Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, 2011.

[267] TM Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[268] D Mitrovic, M Zeppelzauer, and C Breiteneder. Features for content-based audio retrieval. *Advances in Computers*, 78:71–150, 2010.

[269] S Miyamoto. *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Springer, 2010.

[270] KR Müller, S Mika, G Rätsch, K Tsuda, and B Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.

[271] BCJ Moore. *An Introduction to the Psychology of Hearing*. Elsevier, 2004.

[272] F Morrison. *The Art of Modeling Dynamic Systems*. Dover Books, 2008.

[273] MD Mountford. An index of similarity and its application to classificatory problems. In Murphy PW, editor, *Progress in Soil Zoology*, pages 43–50, London, 1962. Butterworths.

[274] SA Mulaik. *Foundations of Factor Analysis*. Chapman and Hall/CRC, 2009.

[275] BB Murdock. The distinctiveness of stimuli. *Psychological Review*, 67(1):16–31, 1960.

[276] NCBI. Genbank website, http://www.ncbi.nlm.nih.gov/genbank/, last visited October 31, 2013.

[277] RE Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

[278] CY Nie, YW Shi, WS Yi, and FQ Cheng. Time-frequency analysis of chaos system based on wigner distribution. *Proceedings Conference on Signal Processing*, 1:396–398, 2004.

[279] NIST. Trec video retrieval evaluation website, http://trecvid.nist.gov/, last visited October 31, 2013.

[280] M Nixon and AS Aguado. *Feature Extraction and Image Processing for Computer Vision*. Academic Press, 2008.

[281] R Nosofsky. Stimulus bias, asymmetric similarity and classification. *Cognitive Psychology*, 23:94–140, 1991.

[282] A Ochiai. Zoogeographic studies on the soleoid fishes found in japan and its neighboring regions. *Bulletin of Japanese Society for Science on Fish*, 22:526–530, 1957.

[283] M Olson and BR Hergenhahn. *Introduction To The Theories Of Learning*. Prentice Hall, 2009.

[284] R Olson and F Attneave. What variables produce similarity grouping? *American Journal of Psychology*, 83:1–21, 1970.

[285] AV Oppenheim and RW Schafer. *Digital Signal Processing*. Prentice Hall, 1975.

[286] SM Pandit and SM Wu. *Time Series and System Analysis with Applications*. Wiley, 1983.

[287] D Parikh and CL Zitnick. The role of features, algorithms and data in visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2328–2335, 2010.

[288] R Parke, E Chew, and C Kyriakakis. Quantitative and visual analysis of the impact of music on perceived emotion of film. *Comput. Entertain.*, 5, 2007.

[289] J Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

[290] K Pearson. On the coefficients of racial likeness. *Biometrica*, 18:105–117, 1926.

[291] K Pearson and D Heron. On theories of association. *Biometrika*, 9:159–315, 1913.

[292] CS Peirce. The numerical measure of the success of predictions. *Science*, 4:453–454, 1884.

[293] M Peruggia. *Discrete Iterated Function Systems*. CRC Press, 1993.

[294] KE Petersen. *Ergodic Theory*. Cambridge University Press, 1990.

[295] P Petta, C Pelachaud, and R Cowie. *Emotion-Oriented Systems: The Humaine Handbook*. Springer, 2011.

[296] JR Pierce. *An Introduction to Information Theory: Symbols, Signals and Noise*. Dover, 1980.

[297] Plato. *Politeia*.

[298] K Pohlmann. *Principles of Digital Audio*. McGraw-Hill/TAB Electronics, 2010.

[299] N Postman. *Technopoly: The Surrender of Culture to Technology*. Vintage, 1993.

[300] WB Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, 2007.

[301] LR Rabiner. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[302] LR Rabiner and RW Schafer. *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[303] SK Reed. *Cognition: Theory and Applications*. Wadsworth Publishing, 2009.

[304] K Reisz and G Millar. *Technique of Film Editing*. Focal Press, 2009.

[305] F Restle. A metric and an ordering on sets. *Psychometrika*, 24:207–220, 1959.

[306] JA Richards and X Jia. *Remote Sensing Digital Image Analysis: An Introduction*. Springer, 2005.

[307] CP Robert and G Casella. *Monte Carlo Statistical Methods*. Springer, 2010.

[308] M Roberts and R Russo. *A Student's Guide to Analysis of Variance*. Routledge, 1999.

[309] C Rocchini. Zernike polynomials algorithm, http://commons.wikimedia.org/wiki/file:zernike_polynomials.png, last visited October 31, 2013.

[310] DJ Rogers and TT Tanimoto. A computer program for classifying plants. *Science*, 132:1115–1118, 1960.

[311] C Rohwer. *Nonlinear Smoothing and Multiresolution Analysis*. Birkhaeuser, 2005.

[312] L Rokach and O Maimon. *Data Mining with Decision Trees: Theory and Applications*. World Scientific Publishing, 2008.

[313] A Rome and M Wilcox. *Multimedia on Symbian OS*. Wiley, 2008.

[314] E Rosch and CB Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7:573–605, 1975.

[315] Y Rubner and C Tomasi. *Perceptual Metrics for Image Database Navigation*. Kluwer, 2001.

[316] Y Rui, TS Huang, and SF Chang. Image retrieval: Past, present and future. *Proceedings of the International Symposium on Multimedia Information Processing*, 1997.

[317] PF Russel and TR Rao. On habitat and association of species of anopheline larvae in south-eastern madras. *Malaria Institute Journal*, 3:153–178, 1940.

[318] RCJ Russell. *Stylometry*. Bookvika Publishing, 2012.

[319] N Sabater, A Almansa, and JM Morel. Meaningful matches in stereovision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):930–942, 2012.

[320] S Santini. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.

[321] S Santini and R Jain. Similarity is a geometer. *Multimedia Tools and Applications*, 5:277–306, 1997.

[322] S Santini and R Jain. Beyond query by example. In *Proceedings of ACM Conference on Multimedia*, pages 345–350, 1998.

[323] Jean-Baptiste Sauvan, Anatole Lécuyer, Fabien Lotte, and Géry Casiez. A performance model of selection techniques for p300-based brain-computer interfaces. pages 2205–2208, 2009.

[324] B Schölkopf and AJ Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[325] C Schmid, R Mohr, and C Bauckhage. Evaluation of interest point detectors. *Computer Vision*, 37(2):151–172, 2000.

[326] Christian Schönauer, Thomas Pintaric, and Hannes Kaufmann. Full body interaction for serious games in motor rehabilitation. pages 4:1–4:8, 2011.

[327] DH Schunk. *Learning Theories: An Educational Perspective*. Addison Wesley, 2011.

[328] WA Scott. Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly*, 19:321–325, 1955.

[329] F Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, 2002.

[330] A Seckel. *The Great Book of Optical Illusions*. Firefly Books, 2009.

[331] JL Semmlow. *Biosignal and Medical Image Processing*. CRC Press, 2008.

[332] M Shah, K Rangarajan, and PS Tsai. Motion trajectories. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):1138–1150, 1993.

[333] CE Shannon and W Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Chicago, 1949.

[334] L Shao, C Shan, J Luo, and M Etoh. *Multimedia Interaction and Intelligent User Interfaces: Principles, Methods and Applications*. Springer, 2010.

[335] R Shepard. Representation of structure in similarity data. *Psychometrika*, 39(4):373–421, 1974.

[336] RN Shepard. Toward a universal law of generalization for psychological science. *Science*, 237:1317–1323, 1987.

[337] GE Shilov. *Linear Algebra*. Dover Publications, Mineola NY, 1977.

[338] TF Shipley. *From Fragments to Objects: Segmentation and Grouping in Vision*. North Holland, 2001.

[339] S Simmons and Z Estes. Individual differences in the perception of similarity and difference. *Cognition*, 108:781–795, 2008.

[340] GG Simpson. Mammals and the nature of continents. *American Journal of Science*, 241:1–31, 1943.

[341] PP Sint. *Similarity Structures and Similarity Measures (in German)*. Austrian Academy of Science Press, Vienna, 1975.

[342] AWM Smeulders, M Worring, S Santini, A Gupta, and R Jain. Content-based image retrieval at the end of the early years. *IEEE Trans Pattern Anal Mach Intell*, 22(12):1349–1380, 2000.

[343] AJ Smola and B Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[344] PHA Sneath and RR Sokal. *Numerical Taxonomy*. W. H. Freeman, San Francisco, 1963.

[345] L Soinov. Bioinformatics and pattern recognition come together. *Journal of Pattern Recognition Research*, 1(1):37–41, 2006.

[346] A Sokal and J Bricmont. *Fashionable Nonsense: Postmodern Intellectuals' Abuse of Science*. Picador, 1999.

[347] RR Sokan and CD Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 38:1409–1438, 1958.

[348] M Sonka, V Hlavac, and R Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing, 1999.

[349] T Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter af Kongelige Danske Videnskabernes Selskab*, 5(4):1–34, 1948.

[350] T Sorgenfrei. Molluscan assemblages from the marine middle miocene of south jutland and their environments. *Danmarks Geologiske Undersøgelse II*, 29:356–503, 1958.

[351] SPSS. Proximities, ftp://ftp.spss.com/pub/spss/statistics/spss/algorithms /proximit.pdf, last visited October 31, 2013.

[352] S Sra, S Nowozin, and SJ Wright. *Optimization for Machine Learning*. MIT Press, 2011.

[353] JF Steffensen. On certain measures of dependence between statistical variables. *Biometrika*, 26:250–255, 1934.

[354] EM Stein and R Shakarchi. *Fourier Analysis: An Introduction*. Princeton University Press, 2003.

[355] S Sternberg. *Dynamical Systems*. Dover Books, 2010.

[356] G Strang. *Introduction to Linear Algebra*. Wellesley Cambridge Press, 2009.

[357] DW Stroock. *An Introduction to Markov Processes*. Springer, 2005.

[358] A Stuart. The estimation and comparison of strengths of association in contingency tables. *Biometrica*, 40:105–110, 1950.

[359] WK Sung. *Algorithms in Bioinformatics*. CRC Press, 2010.

[360] LM Surhone, MT Tennoe, and SF Henssonow. *OpenCV*. Betascript Publishing, 2011.

[361] MJ Swain and DH Ballard. Indexing via color histograms. *International Conference on Computer Vision*, 7:390–393, 1990.

[362] P Symes. *Digital Video Compression*. McGraw-Hill, 2003.

[363] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460 –473, 1978.

[364] H Tang, T Fang, P Du, and P Shi. Intra-dimensional feature diagnosticity in the fuzzy feature contrast model. *Image and Vision Computing Journal*, 26:751–760, 2008.

[365] TT Tanimoto. An elementary mathematical theory of classification and prediction. *IBM Taxonomy Application*, 3:30, 1958.

[366] K Tarwid. Estimating the convergence of ecological niches of species by assessing the likelihood of their getting together in the fishery (in polish). *Ekologia Polska, Serie B*, 6:115–130, 1960.

[367] JS Taylor and N Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[368] P Teetor. *R Cookbook*. O'Reilly, 2011.

[369] JB Tenenbaum, V de Silva, and JC Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[370] JB Tenenbaum and TL Griffiths. Generalization, similarity, and bayesian inference. *Behavioral Brain Science*, 24(4):629–640, 2001.

[371] S Theodoridis and K Koutroumbas. *Pattern Recognition*. Academic Press, 2009.

[372] B Thompson. *Canonical Correlation Analysis: Uses and Interpretation*. Sage Publications, 1984.

[373] W Torgerson. Multidimensional scaling of similarity. *Psychometrika*, 30(4):279–393, 1965.

[374] C Tryfonopoulos, M Koubarakis, and Y Drougas. Information filtering and query indexing for an information retrieval model. *ACM Trans. Inf. Syst.*, 27:10:1–10:47, 2009.

[375] PKC Tse. *Multimedia Information Storage and Retrieval*. IGI Global, 2008.

[376] T Tuytelaars and K Mikolajczyk. Local invariant feature detectors: A survey. *FnT Computer Graphics and Vision*, pages 177–280, 2008.

[377] A Tversky. Features of similarity. *Psychological Review*, 84(4):327–351, 1977.

[378] Waikato University. Weka data miner website, http://www.cs.waikato.ac.nz/ml/weka/, last visited October 31, 2013.

[379] PJ van Laarhoven and EH Aarts. *Simulated Annealing: Theory and Applications*. Springer, 2010.

[380] VN Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[381] VN Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[382] A Vedaldi and B Fulkerson. Vlfeat library, http://www.vlfeat.org/, last visited October 31, 2013.

[383] P Viola and MJ Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[384] J von Neumann. *The Computer and the Brain*. Yale University Press, 1958.

[385] M Wallach. On psychological similarity. *Psychological Review*, 65(2):103–116, 1958.

[386] JZ Wang. *Integrated Region-Based Image Retrieval*. Kluwer, 2001.

[387] M Wang and XS Hua. Active learning in multimedia annotation and retrieval: A survey. *ACM Trans. Intell. Syst. Technol.*, 2:10:1–10:21, 2011.

[388] MJ Warrens. On association coefficients ofr 2x2 tables and properties that do not depend on the marginal distributions. *Psychometrika*, 73(4):777–789, 2008.

[389] MJ Warrens. On the equivalence of cohen's kappa and the hubert-arabie adjusted rand index. *Journal of Classification*, 25:177–183, 2008.

[390] MJ Warrens. On the indeterminancy of resemblance measures for binary data. *Journal of Classification*, 25:125–136, 2008.

[391] J Watkinson. *Introduction to Digital Video*. Focal Press, 2001.

[392] A Watt and F Policarpo. *The Computer Image*. Addison Wesley, 1998.

[393] H Webster. A note on profile similarity. *Psychological Bulletin*, 49:538–539, 1952.

[394] EW Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman and Hall, 1998.

[395] DV Widder. *The Laplace Transform*. Dover Books, 2010.

[396] E Wisniewski. What makes a man similar to a tie? *Cognitive Psychology*, 39:208–238, 1999.

[397] L Wittgenstein. *Philosophical Investigations*. Wiley-Blackwell, 2009.

[398] JK Wu, MS Kankanhalli, JH Lim, and D Hong. *Perspectives on Content-Based Multimedia Systems*. Kluwer, 2000.

[399] Xiao Wu, Wan-Lei Zhao, and Chong-Wah Ngo. Near-duplicate keyframe retrieval with visual keywords and semantic context. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pages 162–169, New York, NY, USA, 2007.

[400] B Wünsche. A survey, classification and analysis of perceptual concepts and their application for the effective visualisation of complex information. In *Proceedings of the 2004 Australasian Symposium on Information Visualisation – Volume 35*, pages 17–24. Australian Computer Society, Inc., 2004.

[401] ZZ Xing, J Pei, and E Keogh. A brief survey on sequence classification. *SIGKDD Explor. Newsl.*, 12:40–48, 2010.

[402] CY Xu and JL Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[403] C Yang. Efficient acoustic index for music retrieval with various degrees of similarity. In *Proceedings of the tenth ACM International Conference on Multimedia*, pages 584–591, New York, NY, USA, 2002. ACM.

[404] GL Yang and LM LeCam. *Asymptotics in Statistics: Some Basic Concepts*. Springer-Verlag, Berlin, 2000.

[405] JY Yang, D Zhang, and JF Lua. Feature fusion: parallel strategy vs. serial strategy. *Pattern Recognition*, 36(6):1369–1381, 2003.

[406] PV Yee and S Haykin. *Regularized Radial Basis Function Networks: Theory and Applications*. Wiley, 2001.

[407] SS Young, PD Scott, and NM Nasrabadi. Object recognition using multilayer hopfield neural network. *IEEE Transactions on Image Processing*, 6(3):357–372, 1997.

[408] L Yu and H Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, 2004.

[409] GU Yule. *An Introduction of the Theory of Statistics*. Charles Griffin and Company, London, 1911.

[410] GU Yule. On the methods of measuring association between two attributes. *Journal of the Royal Statistical Society*, 75:579–642, 1912.

[411] GU Yule and MG Kendall. *An Introduction to the Theory of Statistics*. Charles Griffin and Co., 1950.

[412] W Zeng and HM Zhang. Depth adaptive video stitching. In *Proceedings of the 2009 Eigth IEEE/ACIS International Conference on Computer and Information Science*, pages 1100–1105, Washington, DC, USA, 2009. IEEE Computer Society.

[413] S Zhang and T Sim. Discriminant subspace analysis: A fukunaga-koontz approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1732–1745, 2007.

[414] T Zhang and CCJ Kuo. *Content-Based Audio Classification and Retrieval for Audiovisual Data Parsing*. Kluwer, 2001.

[415] Y Zhang and PI Rockett. A generic multi-dimensional feature extraction method using multiobjective genetic programming. *Evol. Comput.*, 17:89–115, 2009.

[416] W Zhao, R Chellappa, PJ Phillips, and A Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35:399–458, 2003.

[417] KH Zou, A Liu, AI Bandos, and L Ohno-Machado. *Statistical Evaluation of Diagnostic Performance: Topics in ROC Analysis*. Chapman and Hall, 2011.

Index

2d string, 96, 97, 99, 279
3d television, 577

absolute threshold, 428, 437, 439
action potential, 427, 547
action unit, 475
activation function, 550, 551, 554, 558, 559
active contour, 95, 291, 297, 452, 456–458, 564, 565
adaboost, 357, 359, 366, 374, 474
adaptive resonance theory, 554, 556, 557
afferent cell, 546
aggregation, 84, 89, 90, 93, 105, 109, 115, 143, 192, 209, 211, 213, 245, 250, 255, 278–281, 290–294, 296, 297, 301, 372, 380, 381, 391, 401, 434, 446, 466, 550, 555, 558, 560, 565
alignable differences, 541, 542
all-or-none principle, 547
amino acid, 26, 56, 103, 113, 115
analogical reasoning, 521, 531
analogous indicator, 408
analyticity, 323
anchoring problem, 436
angular radial transform, 96, 235, 256, 444
angular turning function, 454
anova, 137
apex, 242, 243
application design, 56, 180, 181, 579
application domain, 156, 185, 196, 405, 442, 460, 465, 574
application types, 468
arbitrary description, 463, 467, 468, 474
area of overlap, 446
arousal, 450, 476
artificial neural network, 545–547, 549, 553, 561, 567
association, 20, 111, 139, 152–154, 166, 278, 279, 298, 319, 385, 491, 495, 523, 524, 541, 542, 559, 575
asymptotically stable, 480
attack time, 67, 68, 204, 448, 476
attractor, 421, 467, 494, 497, 508, 509, 511–514, 548
audio harmonicity, 72, 73
audiovisual, 11, 15, 105, 107, 109, 110, 189, 205, 217, 257, 348, 463, 475
auditory cortex, 428
auditory nerve, 242, 243, 428
auratic, 411, 412
auto-associative, 491
autocorrelation, 69, 71, 72, 77, 78, 90–92, 106, 110, 111, 115, 116, 204, 209–211, 246, 247, 252, 270, 391, 402, 444, 466, 471, 555, 565, 567, 570, 571
autocorrelation matrix, 270
autonomic system, 546
axon, 427, 546, 547

background subtraction, 284–286, 288, 290, 291
backpropagation, 549, 552, 553
backward reasoning, 163, 171
backward selection, 305
bags of features, 80, 153, 278–281, 290, 301, 346, 403, 445, 453, 564, 565
band-limited, 227, 228, 245
bark scale, 243, 245, 261, 428, 565
Barnsley's fern, 467
base classifier, 354
base function, 217, 221–225, 230, 231, 251, 259, 444
bayes rule, 163, 164, 167–169, 172, 174
bayesian classifier, 161, 167, 168, 330
bayesian network, 161, 163, 167, 170–174, 179, 365, 436, 498, 506, 513, 514, 567
bearish behavior, 451
behaviorism, 316
belief score, 67, 68, 70, 106, 258, 381, 443, 451, 472, 480, 484, 538, 570, 579
best fit, 500, 520
bifurcation, 508, 512–514, 516, 517
big picture, v, 14, 16, 37, 39, 42, 50, 52, 54, 57, 107, 113, 139, 184, 201, 202, 205, 207, 208, 213, 215, 217, 262, 273, 278, 396, 402, 404, 468, 469, 471, 477, 573
bigger picture, v, 15, 201, 213, 217, 564
binary classification, 47, 372
binary independence model, 167–170, 330, 331
binary retrieval, 150
binaural hearing, 440
bioinformation, v, 4, 11–13, 17, 20, 22–29, 31, 33, 41, 42, 49, 53, 56, 99, 101, 103, 104, 112, 113, 116, 153, 156, 179, 202, 203, 281, 318, 379, 402, 436, 444, 447, 469, 564
biosignal, v, 4, 9–13, 18, 20, 22–29, 31, 34, 42, 54, 55, 59, 60, 64, 71, 73, 74, 77, 78, 153, 189, 190, 202, 226, 230, 236, 240, 241, 248, 250–252, 257, 261, 280, 281, 286, 311, 402, 443, 444, 447–452, 468, 469, 471, 565, 570
blind spot, 429
blob detection, 270, 297
Boltzmann distribution, 495
Boltzmann machine, 479, 491, 494, 495, 515, 549, 552, 556
boolean retrieval, 23, 352
boosting, 325, 351, 354, 356–358, 363, 366, 367, 393, 396, 397, 468, 473, 476, 502, 503, 514, 558, 559, 566, 567
border locking, 432
Borel set, 509, 510
box counting dimension, 467
brain computer interface, 31, 122
broca region, 428
browsing, 92, 181–183, 191, 194, 197, 206, 207, 329, 330, 343, 347, 357, 375, 397, 439, 589
brute force, 363, 366
bubble kernel, 355
building block, 13, 14, 16, 18, 37, 43, 48, 49, 56, 64, 85, 89, 201, 203, 208, 210, 211, 213, 246, 262, 274, 316, 329, 331, 332, 389, 396, 397, 399–401, 418, 444, 456, 458, 480, 484, 509, 545, 546, 550, 554–557, 560, 561, 568, 573, 574, 586
bullish behavior, 451
butterfly, 450, 451

camera motion, 283, 295, 296, 391, 564,

565
canny edge, 94, 253
canonical correlation analysis, 212, 306, 347, 350, 383–385
cascade correlation, 545, 546, 557, 559, 566, 567
categorization, v, 3, 15, 16, 26, 31, 34, 37–42, 44–46, 48–57, 65, 66, 68, 69, 73, 76, 80, 83, 96, 97, 100, 105, 107–111, 115, 120–122, 133, 135, 137, 139–149, 151, 153, 155–159, 161–164, 166–170, 173–181, 183–189, 191, 193, 194, 196–198, 201–207, 212–218, 252, 258, 261, 262, 273, 278, 279, 287, 289, 290, 298, 299, 301–304, 306, 307, 309, 310, 314–317, 321–323, 325, 327–335, 337, 341–344, 347–354, 356, 357, 359–363, 365–367, 369, 372–374, 376, 382, 385, 386, 389, 390, 392, 393, 395–403, 405, 407, 408, 413, 415, 423, 425, 434, 446, 450, 460–463, 468–474, 477, 486, 487, 491, 497, 498, 503, 504, 507, 508, 513, 517–521, 527, 532, 545, 546, 549–551, 553, 554, 556, 557, 559–561, 563, 564, 567–569, 571–575, 577–579, 595
category, 4–6, 60, 70, 94, 113, 128, 134–136, 139, 141, 202, 211, 212, 248, 322, 323, 325–327, 329, 348–350, 362, 371, 392, 394, 395, 397, 412, 418, 420, 434, 463, 468, 498, 524
causa efficiens, 418
causa finalis, 418
causa formalis, 418
causa materialis, 418
cause and effect, 409
central nervous system, 412, 546, 548
cepstrum, 246
change of aspect, 409, 548
changes of oscillatory activity, 76, 252
channel, 24, 25, 28, 30, 41, 52, 53, 55, 56, 70, 74, 76, 82, 87, 121, 124, 205, 254, 258, 376, 411, 421, 475, 576
chaos theory, 497
Chebyshev polynomial, 236
choice model, 156, 158, 326–328, 337, 346, 434, 435, 520–522, 579
city block distance, 35, 36, 148, 150, 256, 485, 525, 535, 571, 587
class label, 38–42, 44, 46–48, 65, 139, 140, 142, 144, 145, 152, 173, 178, 194, 198, 202, 203, 205–207, 216, 301, 314, 321, 327, 359, 362, 374, 401, 418, 460, 468, 507, 559, 569
classical theory, 323, 324, 446, 634
classification error, 334
classifier fusion, 300
classifier model, 356, 514
cluster analysis, 143, 146, 149–152, 207, 263, 305, 328, 330, 331, 343, 397, 403, 470, 498, 500, 502, 505, 534, 566, 567
co-occurrence measure, 534
coarse representation, 40, 83, 125, 126, 173, 194, 198, 211, 245, 294, 301, 442, 484, 486, 490, 565
coarseness, 88–92, 295, 486
cochlea, 60, 242, 243, 427, 428
codebook vector, 132, 354–356, 486, 493, 515, 552, 558
cognitive-perceptual illusion, 434
cognitive-statistical illusion, 434
coiflet, 233
cold media, 412, 413, 422
color distribution, 84
color histogram, 38, 39, 49, 55, 85–88,

154, 204, 288, 326, 402, 461, 464, 465, 564, 565
color layout, 87, 92, 135, 254, 255
color model, 52, 82, 83, 87, 111, 254, 429, 441, 464, 465
color perception, 81, 437, 441, 462
color structure, 87, 135, 204, 465
color temperature, 441
colorimetry, 441
communication media, 411
competitive learning, 486
complex concept, 322
computer vision, v, 4, 9, 11, 38, 92, 395, 578
concept theory, 322, 325, 392, 435, 444, 446, 500, 520, 566, 567, 572
conceptual atomism, 324
conditional independence, 170, 171, 174
conditional probability distribution, 170–172, 359
confidence, 336, 506, 507, 573
conflict-free system, 8, 491
connotation, 163, 418, 452
consistency axiom, 534
constructivism, 316
content understanding, 252
content-based image retrieval, 3, 79, 472
contextualization, 205, 357, 517, 564, 567, 577
contingency table, 372, 373
contourlet, 236, 237, 256, 444
contracting image transform, 466
control loop, 331, 332, 398
convergence principle, 503, 507
convergent filtering, 479, 480, 482, 487, 495, 556, 557
convex hull, 360, 481
convolution, 15, 29, 48, 64, 69–73, 91, 94, 136, 203, 209–211, 222, 223, 231, 233, 234, 242, 245, 247, 258, 265, 272, 274, 284, 292, 345, 346, 390, 443–447, 449–451, 458, 527, 537, 548, 561, 566, 567, 571, 572
cooling scheme, 364
copy detection, 20, 107
corner detection, 270
corpus, 107, 109, 196, 426
correlogram, 73, 77, 78, 91, 106, 116, 148, 204, 402, 564, 565, 570
cosine ft, 226, 228
cosine spectrum, 252, 253
cosine transform, 87, 224, 225, 228, 230, 240, 246–248, 252–255, 463
critical band, 241, 243, 245, 247, 428, 439, 440
cross validation, 369–372, 374, 375, 393, 404, 495, 566, 567
cross-spectral density, 248, 250, 311, 564
crosscorrelation, 17, 204, 209, 210, 213, 217, 223, 258, 266, 294, 398, 401, 405, 444, 445, 475
crossover, 306, 364, 365
cup with holder, 450, 451, 491
curse of dimensionality, 6, 43, 109, 278, 293, 304, 314, 394, 514, 543
curvature scale space, 455, 564
curvelet, 236, 237
cut-off criterion, 330
cybernetics, 410, 421, 480

data reduction, 132, 158
data cloud, 130, 131
data quality, 88, 369, 370, 385, 569
Daubechies wavelet, 233
decision level fusion, 300
decision rule, 46, 139, 170, 330, 332, 349, 354, 357, 358, 382, 398, 401, 403, 471, 474, 532
decision tree, 39, 144–146, 198, 207, 215, 321, 330, 331, 352, 357, 359, 397, 403, 499, 500, 502, 506, 514, 532, 566, 567, 573

decorrelation, 209–211, 213, 230, 240, 246, 247, 463, 570
deep feature, 322, 446
delta coefficient, 128, 129, 274
dendrite, 426, 546, 547, 550
dendrogram, 115, 149
denotation, 418, 460
density estimation, 90, 91, 165, 168, 169, 192, 319, 320, 329, 331, 332, 335, 351, 366, 392, 501
depth map, 442, 577
depth perception, 433, 434, 556
description space, 42, 122, 134, 398, 476, 498, 500, 510, 569, 579
determinant of the hessian, 269–271
diagnostic feature, 527, 577, 578
diagnosticity, 532
difference of gaussian, 274
digital indicator, 408
dimensionality reduction, 281, 314
Dirac function, 235, 318, 335
directionality, 88, 90–92, 256, 295, 296
discrete transform, 221–225, 228, 230, 231, 233, 240, 241, 390, 445, 452
distance measure, v, 48, 146, 147, 150, 152–154, 157, 197, 223, 293, 326, 328, 345–347, 350, 354, 383, 445, 447, 455, 471, 486, 524–528, 530, 534, 535, 543, 544, 571, 579
distance transform, 255, 256
distinctiveness, 308, 309
divide and conquer, 113, 115, 116, 125, 212, 365, 366, 542, 576
dna, 22, 26, 56, 57, 112, 114, 115, 179
dominance relation, 533
dominant color, 38, 39, 41, 49, 84, 86–88, 111, 204, 464, 564, 565
dominant frequency, 250
dot product, 136, 150, 404, 444, 445, 523–525, 535, 537, 571, 587, 588
dual optimization problem, 341
dual process model, 136, 328, 352, 353, 418, 447, 519, 536–542, 544, 561, 566, 567, 571, 572, 575, 579, 592
dynamic filtering, 16, 301, 477, 480, 482, 484
dynamic programming, 156, 226, 352, 365, 366, 458
dynamic time warping, 23, 53, 57, 113, 154–156, 365, 541, 542, 567, 570
dynamical system, 410, 412, 421, 482, 497, 507–514, 567

ear canal, 428
ear flap, 428
eardrum, 428
early fusion, 300–302, 403
earth mover's distance, 157, 207, 279, 281, 404, 446, 542, 567
ecg, 4, 5, 11, 13, 22, 31–35, 73–75, 77, 249–251, 281
edge detection, 93, 210, 267, 284, 297, 570
edge information, 33, 35, 80, 89, 92–95, 148, 162, 179, 202, 204, 209, 210, 223, 227, 228, 231, 233, 234, 239, 252–254, 256, 257, 263, 265, 267, 270, 271, 274, 275, 277, 289, 291, 312, 391, 394, 429, 430, 433, 434, 447, 453–456, 458, 465, 569
edge map, 94, 95, 240, 253, 267, 457
edgy distribution, 307
edit distance, 113, 155–158, 542, 543
eeg analysis, 11, 312
eeg-based spelling, 471
efferent cell, 546
eidos, 408

eigenvalue, 131, 132, 270, 275, 310, 311, 383, 384
eigenvector, 131, 132, 271, 310, 311, 313, 383, 384
eikon, 408
elastic matching distance, 446, 447
electromyography, 74, 283, 286
emotion recognition, 460, 462, 472, 475, 476
empirical risk, 334–337, 340, 504–506, 525
energy model, 95, 293, 297
energy-based contour model, 456
ensemble method, 145, 325, 397, 401, 403, 566, 567
entropy, 136, 335, 370, 375–381, 422, 423, 440, 489, 490, 528, 553, 566, 567
epistemology, 407–409, 414
ergodic source, 377
ergodic system, 378, 421, 508, 510, 511, 514, 559
ergodic theory, 421
error of first type, 373
error of second type, 373
escape technique, 363
euclidean distance, 72, 115, 148, 150, 151, 255, 288, 335, 338, 354, 530, 532, 558, 559, 587
euclidean geometry, 529
Euler's formula, 223, 225
evaluation process, 193, 196, 369, 370, 372, 377, 393
exhaustive search, 304
expectation maximization, 126, 165–167, 177, 179, 191, 218, 294, 297, 310, 311, 320, 331, 349, 351, 359, 360, 362, 384, 398, 457, 482, 488, 514–517, 559, 560
expected risk, 336, 504–506
explaining away, 172
f1 score, 195, 196, 304, 370, 385, 404, 566, 567, 584
face recognition, 3, 52, 140, 182, 184, 193, 196, 319, 460, 462, 472, 474, 475, 490, 491
facial action coding system, 475, 476
factor analysis, 119, 124, 129–133, 137, 167, 198, 205, 211, 245, 247, 277, 299, 309–313, 347, 403, 447, 490, 566, 569, 579
fade, 287–289
fallout, 194, 373–375
false negatives, 194, 373, 375
false positives, 52, 80, 194, 195, 337, 373, 374, 474
family resemblance, 203, 418
feasibility principle, 503, 507
feature contrast model, 328, 445, 534, 535, 537, 538, 589
feature extraction, v, 37–40, 44, 46, 50, 64, 65, 88, 99, 125, 140, 189, 196, 201, 205, 208, 211, 213, 216, 217, 224, 261, 262, 267, 272, 278, 280, 281, 283, 299, 300, 328, 348, 390, 391, 393, 407, 408, 425, 428, 443, 458, 460, 461, 468–470, 473, 474, 480, 484, 568, 573, 574
feature matrix, 120, 122–126, 128–137, 140, 142, 146, 149, 151–153, 158, 169, 182, 183, 189, 197, 205, 299, 301–307, 309, 310, 312, 314, 329–331, 333, 334, 339, 341, 343, 344, 346, 355, 356, 359, 360, 362, 366, 377, 381, 382, 384–386, 393, 399, 403, 470, 474, 476, 481, 486, 493, 499, 502, 505, 521, 523, 531, 534, 535, 564, 569, 572, 573, 577
feature merging, 119, 120, 205, 299, 403
feature selection, 218, 299, 303–307, 357,

386, 403, 464, 474, 565, 566, 577–579
Fechner law, 437, 438
feed-back neural network, 549, 552
feed-forward neural network, 552, 557, 559
feedback system, 421, 480, 484
flow function, 421, 508, 510–514
forward algorithm, 177–179, 470
forward reasoning, 68, 163, 171, 174
Fourier transform, 35, 60, 222, 225–228, 230, 242, 243, 245–248, 257, 261, 262, 390, 427, 464
Fréchet distance, 446
fractal dimension, 467
ft kernel, 228
fundamental frequency, 5, 10, 34, 62, 69–71, 75–77, 247, 252, 449
fusion of descriptions, 121, 299, 300, 303, 565
fuzzy information retrieval, 351–353
fuzzy method, 24, 328, 351
fuzzy set operator, 538

Gabor transform, 230
Gabor wavelet, 92, 231, 232, 252, 255
game theory, 513
Gartley, 450, 451
gaussian bayes classifier, 307
gaussian filter kernel, 265
gaussian function, 224, 230, 231, 243, 252, 264, 271, 272, 318, 361
gaussian kernel, 345
gaussian mixture, 362, 366, 502, 517
genbank, 56, 102, 113
general training requirements, 503
generalization, 39, 46–48, 145, 146, 155, 236, 278, 290, 316–319, 321, 326–328, 331, 332, 345–347, 359, 371, 372, 375, 395, 399, 401, 521, 522, 524–526, 538–540, 548, 553, 556, 558, 559, 566, 567, 571, 579
genetic algorithm, 56, 208, 213, 306, 355, 364–366, 471, 520, 578
genetic code, 56, 104
gestalt, 47, 218, 273, 278, 394, 434, 454, 460, 465, 527, 568, 569, 578
Gibbs sampling, 91, 165–167, 179, 191, 320, 359, 398
Gini coefficient, 379
gist, 148, 277, 318, 327, 370, 526, 530, 536, 537, 572
global alignment, 112
global optimization, 351, 352, 363, 458
gloh, 277
goal-centered quantization, 485
goggles, 579
goodness of form, 527
gradient, 239, 261, 274–276, 280, 281, 291, 292, 295, 296, 390, 391, 433, 444, 455, 458, 494, 552, 565
grandmother neuron, 316
granularity, 485
graph matching, 24, 156, 542
gray bar illusion, 433
ground truth, 40, 45, 55, 57, 142–146, 148, 152, 157, 161, 163, 167–169, 174, 179, 186, 188, 189, 191–194, 213, 214, 216, 217, 219, 290, 304, 315, 320, 321, 331, 332, 334, 336, 347–351, 357, 359, 365, 370–374, 377, 382–384, 386, 392, 396, 397, 399, 401, 404, 405, 418, 444, 460, 462, 463, 468, 472, 476, 503, 504, 506, 552, 559, 569, 574–576, 578

Haar wavelet, 86, 223, 224, 228, 230, 233, 252, 254
hair cell, 242, 243, 427
Hamming distance, 23, 155, 207, 328,

348, 404, 491, 534, 535, 537, 543, 571, 589
Hamming function, 230, 243, 263
harmonicity, 72, 73, 204, 624
Harris corner detector, 269, 270, 394
Hausdorff dimension, 467
Hausdorff distance, 156, 158, 207, 279, 298, 446, 467, 542, 567
head and shoulders, 450, 451, 461
Hebb rule, 493
hedgers, 206, 207, 218, 219, 316, 325, 329–332, 350, 392, 397–399, 567
Hermann grid, 431, 432
Hesse matrix, 261, 269–273, 275, 280, 281
hidden layer, 173, 550, 551, 558, 560
hidden Markov model, 139, 174–179, 290, 301, 302, 362, 470, 488
hierarchical priors, 172
hill climbing, 363
histogram intersection, 148, 157, 588
homogeneous texture descriptor, 255
homologization, 541
hop size, 65, 66, 262, 288
Hopfield network, 179, 479, 491–496, 515, 549, 552, 556, 557, 560, 565
Horn-Schunck approach, 291
hot media, 412, 413
hough transform, 95, 224, 237, 240, 256, 267, 296
Householder reflection, 310
Hu moments, 96
Huffman coding, 120
hull curve, 440, 448–450
human brain, 33, 80, 83, 408, 414, 425–427, 429, 527, 536, 546, 553–555, 574
human judgment, 7, 142, 326, 375, 395, 406, 461, 462, 532, 571, 573
human perception, 62, 64, 82, 86, 136, 148, 202, 204, 267, 307, 308, 312, 314, 343, 362, 390, 391, 394, 404, 405, 410, 425, 426, 430–432, 434, 436, 442, 461, 530, 565, 577
human similarity, 17, 38, 136, 158, 312, 326, 328, 333, 343, 395, 396, 401, 405, 406, 418, 434, 443, 444, 462, 518, 520, 524–527, 530, 534, 536, 537, 540, 579
Hume's law, 409
hybrid fusion, 300, 301, 303, 566

iconic relationship, 417–419
ideogram, 414, 415
image theory, 408, 416, 520
independent component analysis, 60, 309, 311, 312
indexical, 416–418, 420
information filtering, v, 16, 38, 39, 42, 48, 50, 55, 119, 120, 122, 123, 125, 129, 133, 137, 139, 140, 181, 184, 188, 201–203, 205, 218, 245, 248, 259, 298, 299, 302, 347, 369, 375, 382, 386, 390–393, 403, 460, 473, 479, 484, 491, 504, 554, 556, 557, 563, 565, 568, 569, 578, 594
information gain, 378
information media, 411
information theory, 375, 376, 378, 405, 407, 410, 412, 419, 421, 423, 511
information visualization, 134, 189
instrumentation, 25, 61, 70
integral feature, 527
integral image, 473, 474
integral stimuli, 72, 223, 303, 445, 446, 527, 530, 537, 540, 542
integral transform, 15, 35, 92, 210, 217, 221, 222, 224, 291
interest point, 23, 158, 218, 261, 264,

266, 267, 269–278, 281, 284, 291, 297, 390, 394, 395, 402, 421, 430, 434, 453, 454, 466, 475, 484, 564, 565, 568–571, 578
interestingness measure, 309, 375, 379–382, 403, 412, 423, 489, 490, 566, 567
internal energy, 457, 458
interneuron, 546
interpretation, 4–6, 8, 9, 43, 168, 174, 205, 209, 211, 213, 221, 223, 224, 240, 246, 274, 275, 284, 317, 346, 372, 374, 421, 444, 445, 463, 555, 557, 564, 640
interval scale, 33, 123, 124, 523, 539
isomap, 309, 312, 313, 565, 566, 579
iterated function system, 114, 115, 179, 466, 467, 469, 480, 511, 542, 544
iterative refinement, 40, 184, 185, 191, 193, 207, 302, 331
Itten color wheel, 464

Jacobi matrix, 269, 511
just noticeable difference, 437, 439

k complex, 76, 77, 252, 281
k-means, 146, 151, 152, 166, 197, 206, 300, 320, 330, 331, 351, 353–355, 397, 398, 403, 462, 484, 486, 487, 499, 500, 502, 505, 506, 566, 567
k-nn, 146, 152, 153, 166, 197, 207, 330, 331, 396, 397, 403, 462, 476, 503, 505, 514, 566, 567
Kaiser criterion, 132
Kalman filter, 17, 297, 301, 376, 479, 481, 482, 487–491, 495, 496, 514, 515, 517, 566, 569
kalmanface, 490
kansei features, 464, 465
kernel function, 132, 333, 342–348, 350, 382, 386, 393, 396, 403, 464, 468, 521, 571, 579
kernel mapping, 344, 539
keypoint, 276, 277, 453, 454
Kolmogorov complexity, 543
Kravchuk polynomial, 236
Krumhansl model, 326, 327, 343, 355, 505, 525, 575
Kullback Leibler divergence, 309, 379, 380, 588

Lagrange approach, 341, 342, 348, 383, 384
language understanding, 428
Laplace operator, 93, 266, 274, 430
Laplace transform, 221–225, 230, 520
laplacian of gaussian, 272, 274, 394
late fusion, 300–303
latent semantic indexing, 311
lateral inhibition, 432
law of great numbers, 370, 435
law of resemblance, 409
law of small numbers, 370
learning ability, 503
learning procedure, 23, 45, 359, 398, 552, 553, 574
learning rate, 338, 354, 364, 486, 495, 552, 553
least squares, 72
Lebesgue measure, 509–511
Levenshtein metric, 155, 156, 543
limited base, 221, 224, 230, 254
linear discriminant analysis, 212, 334, 347–350, 382, 383, 502, 566, 567
linear kernel, 345, 403
linear predictive coding, 71–73, 78, 91, 204, 247, 295, 471, 565, 570
local alignment, 112, 156, 281
local feature, 64, 87, 154, 217, 234, 261, 264, 266, 267, 272, 273, 277,

279–281, 283, 296, 390, 391, 395, 453, 460, 537, 578
local optimum, 100, 166, 306, 362, 363, 365, 480, 504
localization, 84, 85, 87–90, 115, 203, 208, 209, 211–213, 217, 234, 243, 245, 246, 261–264, 274, 275, 278, 284, 287, 290, 292, 294, 301, 401, 432, 440, 460, 555
logistic map, 513, 559
long-time memory, 548, 557
loss function, 334–337, 340, 348, 357, 392
loss variable, 358
Lotka-Volterra equations, 516
loudness, 43, 62–68, 70, 77, 86, 241, 243, 245, 247, 428, 438–440, 464, 556, 569
Lucas Kanade approach, 391

machine learning, vi, 3–7, 10–18, 20, 22–24, 26, 35–38, 44–46, 113, 139, 142, 143, 145, 184, 189, 218, 257, 315–320, 322, 325, 328, 333–335, 356, 359, 363, 392, 393, 396, 461, 497, 503, 505, 507, 521, 557, 566–568, 571–573, 575, 578
macro process, 143, 144, 146, 153, 154, 158, 206, 315, 326, 331, 337, 369, 392, 405, 500, 571, 572
macroblock, 94, 284, 290, 292, 293, 391
Mahalanobis distance, 326, 349, 523, 587
Mallows distance, 157, 454, 523, 542
man is the measure, 406, 431, 462, 472, 521, 545, 553, 568
mapping function, 344–347
Markov process, 165, 174–176, 188, 376, 378, 403, 488, 492
Markov random field, 91, 170, 174, 179
masking, 243, 440, 441, 462
massenkunst, 411, 415
matlab, 189, 190, 568
maximal stable extremal regions, 272, 273
maximum margin, 339
McCulloch-Pitts neuron, 492
mean shift, 127, 489
measure theory, 508, 509
measurement process, 150, 153, 156, 327, 328, 346, 369, 398, 511, 523, 524, 530, 534
measurement update, 488, 517
media object, 14, 19, 26–30, 34–38, 40–49, 57, 61, 65, 67, 79–81, 83–87, 97, 99–101, 109, 115–117, 119, 121–126, 133–136, 139, 140, 142, 143, 149, 151, 152, 154, 158, 168, 178, 182, 183, 186, 188, 194–197, 201–210, 221–224, 245, 261, 262, 264, 268, 271–273, 276, 277, 281, 293, 317, 327, 352, 353, 362, 379, 382, 391, 408, 411, 412, 415, 418, 419, 423, 443, 444, 452, 459, 460, 468, 474, 476, 490, 533, 539, 554, 555, 569, 576
media theory, 16, 379, 405, 407, 408, 410–412, 414, 415, 419, 421, 423, 567
media understanding of media understanding, 111, 182, 184, 186, 207, 213, 279, 287, 328, 470
Meixner polynomial, 236
mel frequency cepstral coefficient, 246–248, 252, 258, 295, 402, 463, 469, 470, 476, 564, 565, 570
mel scale, 64, 246, 247, 565
membership value, 360, 361
mental representation, 316–318, 320, 408, 554

mental theory, 317, 319–322, 324
Mercer's theorem, 344
metric axioms, 147, 326, 462, 524, 527, 528, 533
mexican hat, 231, 232, 251
Meyer wavelet, 231, 232, 252
micro process, 143, 146, 153, 154, 156, 158, 196–198, 206, 298, 315, 326, 328, 392, 395, 398, 405, 446, 498, 500, 501, 518, 519, 522, 541, 544, 571, 572
minimum risk metric, 337
Minkowski distance, 148, 154, 207, 256, 326, 444, 486, 523–526, 532, 537, 559, 571, 575, 587
misconceptions of chance, 435
mixture model, 351, 359–364, 393, 397, 399, 434, 488, 498, 506, 514, 515, 517, 532, 566, 567
mobile media understanding, 182, 186, 187
model estimation, 331, 332, 398, 400, 498
monotone proximity structure, 533, 534
Monte Carlo, 165, 166
morlet wavelet, 231, 232, 252
mother wavelet, 231, 232, 234, 251, 391
motif finding, 112, 115
motion activity, 284, 286, 288, 290, 291, 295, 391, 564
motion description, 256, 283–285, 287, 290, 295, 296, 298, 402, 442
motion trajectory, 283, 295–298, 442, 564, 565
motion vector, 284, 286, 287, 290–292, 295–297
motor cortex, 426
mpeg-7, 67, 72, 84, 86–88, 92, 94, 96, 135, 204, 205, 235, 248, 254–256, 286, 448, 453, 455, 465
mrna, 114
multi-dimensional scaling, 136, 354, 356, 470
multi-resolution analysis, 233, 254, 257, 265, 266, 391, 564
multimedia, v, 3–13, 35–40, 57, 78, 117, 121, 132, 133, 202, 203, 205, 563, 565–570, 572–574, 576, 577, 579
multiple instance learning, 321
music genre classification, 3, 7, 140, 329, 441, 462, 469–471, 487
music retrieval, 4, 186
mutation, 115, 157, 306, 364, 365

n-gram, 107, 109–111, 115, 116, 302, 311, 346, 402, 487, 565
nabla operator, 269
Needleman-Wunsch algorithm, 113, 156, 542
negentropy, 378, 379, 422, 423
neighborhood, 26, 29, 67, 90, 94, 100, 103, 111, 127, 128, 152, 209, 217, 218, 222, 239, 265, 266, 269–271, 273, 274, 276–278, 281, 284, 291–294, 296, 313, 318, 322, 325, 354–356, 378, 391, 395, 429, 430, 487, 512, 565
neighborhood kernel, 354–356
neoclassical theory, 323, 324
neurotransmitter, 242, 427, 547, 555, 561
nominal scale, 25, 123, 522, 540
norm theory, 319, 531, 532, 565, 569
normal distribution, 102, 133, 135–137, 166, 167, 318
normalization, 44, 65, 124, 125, 131, 132, 205, 299, 307, 309, 346, 351, 379, 403, 484, 487, 524, 540, 565, 566
nucleotide, 56, 57, 113, 115
nucleus, 547, 550
Nyquist law, 486

object contour detection, 262
object energy, 457, 458
object recognition, 12, 110, 256, 264, 267, 286, 297
octave, 264
off bipolar cell, 430
off ganglia cell, 430, 554
on bipolar cell, 430
on ganglia cell, 430
one class neighbor machine, 348
open set, 508–510
opencv, 189
operations research, 126, 305, 363, 372
optic chiasm, 429
optic nerve, 81, 429
optic radiation, 429, 430
optical flow, 283, 290–298, 391, 395, 402, 403, 565
orbit, 429, 508, 511, 512, 514
ordinal scale, 123, 522, 523
organ of Corti, 242
orientation histogram, 277
ossicles, 428
over-narrowing, 462, 463, 472
overfitting, 45, 143–145, 157, 165, 189, 315, 324, 330–332, 336, 357, 359, 392, 399, 456, 463, 474, 502, 503, 506, 560, 567, 572

p300 detection, 20, 140, 251, 280
Paice model, 353
parallel postulate, 529
parameter interpolation, 447, 449
parametric family, 359, 360
parametric transform, 221, 224, 237, 238, 240, 256
parts of speech, 109
pattern difference, 328, 589
pattern recognition, 4, 11, 204, 319, 329, 335, 340, 546
peak detection, 77, 204, 281, 296, 471, 570
perceptron, 337, 338, 342, 345, 406, 549, 551, 561, 566, 567
perceptual linear prediction, 247, 248, 252, 258, 471, 564
perceptual load, 427
perceptual-physiological illusion, 432
perfect similarity, 493, 525
periodicity, 34, 77, 78
periodogram, 250, 252, 257
peripheral nervous system, 546
phase correlation, 291, 294
phase space, 510
pictogram, 414
pitch detection, 247
pitch histogram, 71–73, 78
Plato's problem, 323, 444
point of stability, 481
point of subjective equality, 437
point-based template, 453
polynomial kernel, 346
polysemy, 5, 43, 102, 108, 125, 332, 353, 362, 413, 417–419, 421, 458, 476, 544
Ponzo illusion, 433
poor in details, 413
population genetics, 113, 116
pragmatic relationship, 416–418
pre-attentive perception, 527
16, 195
precision, 89, 125, 194–196, 262, 301, 302, 304, 357, 370, 495, 567, 584
predicate-based similarity measurement, 408, 519, 530, 533, 536, 538
preimage, 510
primitive concept, 322
probabilistic inference, 161, 163, 179, 290, 331, 332, 349, 359, 362, 403, 476, 542
probabilistic model, 161, 162, 167, 169, 170, 290, 332

probably approximately correct learning, 507, 567
product rule, 326
projective space, 529, 530
proto-predicate, 166, 173, 216
prototype theory, 323–325, 327, 572
psychoacoustics, 62, 246, 252, 405, 426, 439–441
psychological similarity, 12, 536, 541, 559, 575, 579
psychology of vision, 426, 441
psychophysical model, 425, 436
psychophysics, 16, 405, 425, 426, 436, 439, 441, 442, 564, 565, 568–571
pulse, 5, 31, 74, 75, 248, 251, 427
pure color, 464
put the human in the loop, 45, 214, 461, 469, 568, 572
pyramidal coding, 231, 233, 234, 237

quadratic programming, 342
qualitative measurement, 523, 536, 537, 539, 540
quantitative measurement, 523, 537, 539, 540
quantization error, 355, 356, 485, 486, 552
quantization model, 538, 539
quantized weights, 500
query acceleration, 181, 196, 197
query by example, 40
query by humming, 20
query by sketch, 40
query by text, 40

R statistics package, 189, 568
r-tree, 198
radial basis function, 397, 498, 545, 546, 557, 559, 566, 567
radon transform, 224, 237–240, 466
random classification, 330
random fern, 168
random forest, 145, 146, 331, 397, 463, 566, 567
random step function, 468
randomization, 165, 166
rapidminer, 306
ratio scale, 124, 523
recall, 194–196, 304, 357, 370, 373–375, 566, 567
receiver operating characteristic, 372, 393, 404, 566, 567
receptive field, 75, 559
receptor cell, 81, 427, 430
rectangle feature, 473, 474, 476
rectangular segmentation, 262
recurrent network, 492, 549, 552, 556, 560
recursive search, 365, 366
reduction, 50, 52, 110, 119, 126, 132, 140, 158, 209–211, 213, 228, 230, 262, 263, 277, 281, 306, 314, 364, 395, 415, 466, 555, 577, 627, 628
redundancy, 20, 26, 27, 48, 116, 117, 119, 120, 123, 125, 129–132, 139, 205, 210, 213, 262, 265, 284, 299, 302, 306, 307, 309, 311–314, 335, 378, 379, 391, 415, 456, 465, 564–566, 569, 571, 579
reference, vi, 14, 53, 108, 111, 142, 143, 146, 148–152, 156, 161, 163, 166, 168, 173, 182, 188, 192, 196–198, 206, 213–217, 298, 315, 319–321, 325, 327, 330, 331, 354–356, 359, 363, 384, 397, 398, 403, 405, 428, 434, 435, 445, 484–487, 491, 498–501, 503, 505, 515, 525, 526, 537, 540, 554, 563
region merging, 263
region splitting, 263, 264

regression, 22, 72, 78, 102, 105, 125, 128, 205, 218, 289, 291, 292, 297, 318, 319, 329, 334, 335, 337, 338, 340, 342, 347–350, 391, 392, 403, 406, 435, 491, 535, 551
regularity, 88–92, 295, 450
reinforcement learning, 320, 321, 352, 365, 366
relevance feedback, 40, 184–186, 191, 207, 211, 213, 331
relevance vector machine, 349
representativeness, 434, 435, 503, 569
rest position, 511
result set, 40, 149, 150, 152, 191, 195, 197, 330
retina, 429
retrieval, 3–5, 7, 10–13, 40, 50, 79, 80, 106, 107, 140, 141, 149, 150, 152, 153, 155, 156, 158, 167–169, 182, 183, 185, 190, 194, 195, 197, 206, 207, 329, 330, 338, 343, 352, 353, 357, 372, 375, 397, 439, 470, 472, 491, 546, 563, 565–570, 572–574, 576, 577, 579
retrieval status value, 168
reversible figure, 409
reward, 57, 195, 321, 365–367
rhythm, 34, 42, 60–62, 64, 70–72, 76–78, 89–91, 106, 120, 227, 241, 245, 247, 248, 252, 280, 422, 448, 449, 556
rich in details, 413
ridge detection, 271, 275, 465
ridgelet, 236, 256
risk functional, 334, 335
risk minimization, v, 16, 45, 205, 217, 218, 333–340, 344, 347, 349, 392, 396, 474, 567
roc analysis, 372–375
roc curve, 372, 374
roughness, 439, 440, 449

saccadic seeing, 81, 267, 429, 432
scalable color, 86
scalar quantization, 479, 484, 495
scale of similarity measurement, 445, 537
scale space, 234, 237, 261, 262, 264–266, 270, 272–274, 277, 280, 281, 290, 390, 391, 452, 454–456, 564
scale-invariant feature transform, 274–279, 291, 297, 402
scene classification, 474, 475
scene grouping, 472
search algorithm, 307, 363, 364
search set complexity, 435
self-organizing map, 151, 351, 353–356, 359, 363, 364, 397, 398, 403, 462, 470, 486, 487, 499, 502, 505, 506, 514–516, 549, 552, 557–559, 566, 567, 579
self-similarity, 26, 73, 466, 467, 544
semantic description, 17, 68, 207, 311, 391, 459, 460, 536
semantic enrichment, 41, 42, 301, 460, 461
semantic gap, 5, 45, 46, 52, 57, 62, 109, 110, 128, 133, 174, 176, 179, 185, 208, 216, 221, 224, 240, 259, 281, 298, 314, 332, 366, 394, 400, 442, 461, 476, 573
semantic scale, 459–463, 566, 568
semantic wavelet mother, 393
semantics, 88, 101, 107–109, 111, 153, 213, 272, 278, 287, 311, 315, 322, 359, 394, 400, 405, 423, 459–461, 463, 465, 468, 469, 496, 530, 544, 561, 564, 569, 576
semi-automatic media understanding, 469

semicircular canals, 428
semiotic relationship, 408, 419
semiotics, 99, 405, 407, 408, 415, 418–420, 472, 567
sensory memory, 547
separable, 223, 228, 234, 236, 268, 302, 303, 343, 403, 445, 446, 451, 505, 532, 536, 537, 540–542
separators, 206, 207, 218, 316, 324, 325, 329–332, 339, 350, 392, 397–399, 561, 567
sequence similarity, 112, 113, 156
shape context, 455
shape descriptor, 96, 236, 256, 453, 455
shape retrieval, 158
sharpness, 247, 248, 258, 439, 440
short time energy, 66, 69, 77, 105, 124, 204, 209
short-time memory, 547, 548
sigmoid curve, 345, 481
signal smoothing, 448, 451
signal detection, 7, 20, 372
signal noise ratio, 350, 380–383, 385
signal processing, vi, 3–7, 9–18, 20, 22–24, 26, 31, 35, 37, 38, 75, 78, 184, 188, 189, 203, 204, 218, 263, 426, 520, 528, 573
signified, 408, 416–419, 421, 423
signifier, 416–423
silence descriptor, 464
similarity assessment, 38, 317, 520
similarity function, 156, 333, 344, 347, 352, 400, 530, 538, 539
similarity measure, 48, 54, 223, 279, 325, 326, 328, 344, 346, 348, 445, 447, 465, 486, 491, 496, 511, 520, 524–526, 531, 533–535, 539, 542, 561, 567, 571, 587–591
similarity meta model, 346, 447, 449, 452, 454, 542, 566, 567, 575
similarity perception, 17, 136, 158, 308, 328, 333, 343, 418, 434, 444, 518–520, 524, 526–528, 530, 531, 536
simulated annealing, 305, 364, 366, 495
single neuron doctrine, 316
singular value, 309, 310, 565, 566
singular vector, 310
skewness, 96, 127
slack variable, 341–343, 348, 551
slow cortical potential, 76, 77, 252, 280
Smith-Waterman algorithm, 156
smoothing, 192, 241, 243, 245, 250, 257, 262, 263, 272, 281, 297, 299, 302, 307, 384, 391, 392, 448, 450–452, 454, 458, 491, 638, 641
Sobel operator, 93, 94, 210, 291
soft margin, 341, 343
soma, 546, 547, 550
somatic nervous system, 426, 546
sone transform, 64, 245, 247, 251, 564
sound localization, 440
sound pressure level, 62, 428, 439, 441, 569
source separation, 126, 218, 248, 311, 312
sparse representation, 125–127, 130, 132, 133
spectral feature, v, 73, 87, 217, 221, 224, 240, 241, 245, 247, 248, 254, 257, 259
spectral flux, 247, 248
spectral method, 259, 291
spectrogram, 247, 251, 252
speech recognition, 3, 4, 11, 12, 52, 70, 108, 139, 177, 179, 186, 245, 246, 405, 460, 462, 463, 469, 470
spike response model, 546, 554, 566, 567
spike train, 430, 547, 548, 554
spline, 456–458

split and merge, 263, 264
spread parameter, 559
stability criterion, 382, 383, 480
start codon, 56
statistical mean, 297, 301, 338, 371
statistical moment, 47, 55, 66, 67, 70, 77, 90–93, 95, 96, 105, 111, 115, 116, 124–127, 135, 204, 205, 209, 294, 295, 375, 448, 471, 489, 525, 570
steady-state visual evoked potential, 75
Stephen's exponent, 438, 439
Stephen's power law, 438, 439
stereo vision, 429, 442
stereopsis, 442
stock analysis, 3, 11, 13, 15, 31, 101, 102, 105, 106, 226, 257, 258, 281, 312, 448
stock data, 22, 23, 25, 26, 28, 31, 33, 99, 102–105, 111, 112, 116, 128, 203, 204, 233, 236, 241, 257, 258, 261, 319, 329, 403, 436, 443, 444, 447, 448, 450–452, 469, 491, 565
stop codon, 22, 56
strange attractor, 497, 508, 512, 515
string kernel, 344, 346
string subsequence kernel, 346
structural alignment, 4, 6, 12, 57, 112–114, 156, 346, 365, 452, 469, 519, 520, 541, 542, 544, 564, 565, 567, 570, 579
stylometry, 405, 471, 472
subsemantic, 431, 461–463, 476, 521, 568
subspace analysis, 312
supersemantic, 406, 431, 436, 461–463, 476, 500, 501, 521, 524, 526, 530, 568, 569
supervised learning, 320–322, 354, 552, 640
support vector machine, 333, 334, 336, 337, 339–345, 347–350, 366, 372, 382, 397, 403, 499, 500, 502, 503, 506, 514, 549, 551, 566, 567
supremal distance, 523, 524
surf, 277
surface feature, 322, 446, 531
susan approach, 271
symbolic, 26, 33, 41, 44, 97, 99–102, 104–106, 111, 112, 116, 120, 123, 139, 141, 143–145, 155, 156, 188, 190, 203, 211, 217, 222, 261, 280, 281, 344, 411, 416–420, 444, 447, 452, 564, 565, 570, 577
synapse, 242, 426, 546, 547, 550
syntactic relationship, 408, 417

t-periodic, 511, 514, 515
t-test, 136, 304
Tamura features, 91
taxa, 112, 115, 530
taxonomic judgment, 537
taxonomic thinking, 531, 532, 535–537, 539, 572
taxonomy, 112, 322, 331, 446, 530
template matching, 13, 17, 22, 204, 210, 251, 266, 270, 274, 292, 297, 307, 390, 442–452, 457, 458, 463, 466, 477, 524, 564, 565, 567, 570, 571
template metric, 446, 467
template preparation, 448
template representation, 444, 452, 454–456
tempo, 60, 61
temporal segmentation, 283, 287, 289, 290, 475, 564
test set, 143, 146, 193, 206, 315, 334, 336, 371, 372
text information retrieval, v, 11, 13, 168, 301

text summarization, 111, 204, 570
texture, 10, 23, 42, 79, 83, 84, 87–92, 97, 106, 204, 278, 287, 295, 300, 391, 466, 564, 565
thalamus, 429, 430
the winner takes it all, 486
thematic judgment, 446, 536, 537, 540
thematic taxonomic bridge, 419, 529
thematic thinking, 522, 531, 532, 535, 536, 539, 561, 572
theory theory, 324, 393, 460, 500
thermal equilibrium, 495
three stimuli theory, 26, 82, 429
threshold of discrimination, 437
threshold of hearing, 62, 441
thresholding, 263, 264, 332, 404
timbre, 61, 62, 64, 70, 73, 241, 245, 247, 248, 255, 258, 556
time update, 488, 489
time-limited, 62, 223, 230, 231, 233, 281
tip link, 242, 427
top down strategy, 365
Toronto school, 407, 410, 412
training set, 142, 152, 165, 166, 183, 206, 334, 343, 347, 371, 372, 392, 506, 553, 559
transducer, 376–379, 384
transduction, 321, 407
transformational similarity, 511, 519, 531, 541–544, 567, 579
transitivity, 534
trendline, 105, 106
triad, 526, 527, 540
triangle inequality, 197, 326, 462, 528–530, 533
trigram, 301, 302
true negatives, 194, 373
true positives, 194, 195, 373–375
twin comparison, 289, 565
two-rectangle, 474
typicality, 323, 434, 435
unsupervised learning, 320, 322, 354
underfitting, 324, 330, 342, 399
uniform distribution, 133, 135, 250, 307, 335, 377–379, 385, 422, 466
unimedium, 576, 577
unit distribution, 136, 379

valence, 47, 113, 172, 201, 222, 417, 422, 476, 535, 536, 571
vc dimension, 505–507, 553
vector product, 523–525
vector quantization, 356, 479, 481, 482, 484, 486, 487, 491, 495, 496, 514–516, 565, 566
vector space, 139, 146, 149, 150, 152, 179, 206, 326, 330, 351, 397, 399, 462, 500, 505, 506, 514, 522, 566, 567
video compression, 230, 283, 284, 290
video editing, 287
video surveillance, 3, 6, 20, 146, 442, 472, 475
view-based template, 453
Viola-jones approach, 472, 474, 476
violence detection, 294, 462
visual interpretation, 374
visual acuity, 441, 442
visual cognition, 284
visual cortex, 284, 426, 429, 430, 432
visual data, 13, 15, 26, 59, 79, 83, 84, 87, 97, 99, 108, 133, 148, 157, 189, 190, 202, 204, 235, 237, 241, 261, 267, 273, 279, 281, 285, 290, 432, 439, 442, 443, 448, 460, 467, 472, 475, 527, 544, 577
visual keywords, 79–81, 84, 204, 262, 278, 444, 452, 564
visual retrieval, v, 169
visual template, 13, 446, 452
Viterbi algorithm, 177, 178

Waller-Kraft model, 353
Walsh Hadamard transform, 257
watershed segmentation, 263, 264
wavelet decomposition, 253, 254, 258, 277
wavelet transform, 228, 230, 231, 233–237, 248, 252, 257, 284
weak classifier, 144, 354, 357, 374, 397, 398
Weber law, 437, 438
weighted average, 102, 375
weighting function, 127, 354, 428
weka, 135, 189, 213, 306, 331, 374, 568, 594
Wernicke region, 428
window smoothing, 243, 245, 250, 257
window size, 65, 66, 70–73, 77, 78, 91, 115, 209, 243, 245, 247, 252, 262
windowing, 77, 84, 91, 114, 203, 204, 208, 230, 245, 255, 261–263, 570
winning node, 354–356, 486, 487, 491, 556–559, 579
wipe, 287–290, 353
world information, 214, 369–372, 375, 382, 397, 398, 404, 460, 484

z transform, 230, 251
Zernike polynomial, 235, 256
zero crossing, 34, 53–57, 67, 69, 71, 75, 77, 94, 100, 105, 115, 122, 177, 204, 245, 255, 267, 268, 403, 461, 564, 565
zigzag scan, 254, 255, 455, 490

Abstract

Multimedia information retrieval: That is the desire to make computers see, hear and understand like humans do. Is it possible to give perception to machines, to make them understand facial expressions, hummed melodies, stock charts and ECG curves? If yes, the computer would become an even more valuable companion in business and private life. Think of the possibilities in, for example, healthcare, home security, online customer support or market analysis. This book explains what is possible in multimedia information retrieval today and what is not. We introduce the basic concepts, explain why the first step is always summarization and the second classification, which is essentially applying human understanding of some context on the summary. We group and discuss the various methods that have been proposed for the summarization of audio, visual and other media information. In classification, we build on today's psychological understanding of human cognition. Successfully, we transfer concepts of human similarity perception on machine classification. We cluster machine learning methods by their approach, model and process. On top of that, we link back from the state of the art methods of multimedia information retrieval to human cognition: We propose artificial neural structures for the building blocks of media summarization and classification. The result is a balanced introduction into the field that starts from graduate IT knowledge and ends at the current frontiers of multimedia research.

About the Author

Horst Eidenberger is associate professor of applied computer science at the Vienna University of Technology. He received his Doctor degree in 2000 from the University of Vienna and finished his *Habilitation* in 2005. He has published several books and more than 70 scientific papers in journals and at international conferences. His research interests include automated content understanding, machine learning and signal processing.