

Michael M. Richter · Sheuli Paul
Veton Këpuska · Marius Silaghi

Signal Processing and Machine Learning with Applications

Signal Processing and Machine Learning with Applications

Michael M. Richter •
Sheuli Paul • Veton Këpuska • Marius Silaghi

Signal Processing and Machine Learning with Applications



Springer

Michael M. Richter (deceased)
Electrical and Computer Engineering
University of Kaiserslautern
Kaiserslautern, Germany

Veton Këpuska
Florida Institute of Technology
Melbourne, FL, USA

Sheuli Paul
FB Elektrotechnik & Informationstechnik
Technische Universität Kaiserslautern
Kaiserslautern, Germany

Marius Silaghi
Florida Institute of Technology
Melbourne, FL, USA

ISBN 978-3-319-45371-2 ISBN 978-3-319-45372-9 (eBook)
<https://doi.org/10.1007/978-3-319-45372-9>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Learning has always a goal. For this reason, for a given process, the question is what can we learn about it? Despite the diversities in processes, one can find out the common principles underlying them. Therefore, we discuss stochastic processes from a general point of view, first. As for practical applications, we will consider them through the lens of historical developments by considering the following aspects:

1. The invention of numbers and number systems is our starting point. From numbers and their representations, humans invented the theory of signals and systems. Signals do not occur isolated but within a process.
2. The naturally occurring processes are not deterministic but rather stochastic. Hence, the underlying states are often unknown. That means those processes are mostly described with hidden stochastic models (HSM).
3. Signal processes contain information. However, this information is not directly visible or understandable to humans. The information is hidden in properties of the processes. Properties of signal processes are mostly not directly measurable, and we have to find them out, at least approximately.
4. Information about the unknown elements is necessary. One needs methods to capture them, and here lie the major challenges. To address these challenges, we concentrate on machine learning techniques.

Real-world problems are conceptualized, explained in applied domains. The book is intended for persons interested in the techniques and foundations related to signal processing. As mentioned previously, signals occur within processes and are considered not to be fully observable and predictable; hence, they are ruled by probabilities. Therefore, they are termed stochastic processes. We concentrate on properties of signals that are useful for practical applications. One has rather to manipulate the information available about processes using computations. Doing this properly is a major challenge. The computations needed depend very much on the type of signals and the application domain.

The development of practical applications was foremost in our mind when writing this book. principles where the problems and differences are exhibited

in their realizations. today one is deploying robotic systems to perform various tasks. This in turn requires solving communication problems between humans and machines. communicate. Humans and machines can communicate, for instance, using speech, images, and videos. These all have in common that a physical exchange has to take place. At the lowest level, it requires use of elementary signals that could be re-interpreted on higher levels of abstraction. We describe our methods as organized in a machine learning process. There, one records many examples of signals in order to use them when applying machine learning techniques. The techniques involve several transformations. A major difficulty is to find out which transformation one should apply in which situation.

- First, audio typically uses cascaded IIR filters to obtain high performance with minimal latency. Noting that each filtering stage propagates the errors of previous stages, a high degree of precision in both the signal and coefficients is required in order to minimize the effects of these propagated errors.
- Second, the signal accuracy must be maintained, even as its magnitude approaches zero (this is necessary because of the sensitivity of the human ear).

The signals occur in sets, and these sets contain information and have special structures. Important structures are governed by logic. Often, the signals arise from some events and are not part of a systematic communication. Sometimes they are represented in a linear ordering. However, typically they are not ordered and are often described through general stochastic processes.

Humans are interested in the information contained in any process, be it discrete (e.g., digital) or continuous and modeled by deterministic or stochastic processes. One can employ machine learning methods to find the information governing stochastic processes. Here, one needs to employ some combinatorial, probabilistic, and learning algorithms. Learning in particular is needed if the signal sets are complex, or the signals are corrupted by noise. is for efficiency reasons very restricted by the large number of occurring signals and by the noise. explore systematically these topics. Therefore, we show their usefulness in a number of widely reaching application areas. We intend to represent the whole picture of signal processing instead of restricting it to special aspects. Then we look at recognition problems. For both, machine learning provides a wealth of important techniques. Machine learning is a wide area. In this book, we discuss aspects of it related to signal processes. These are mainly supervised and unsupervised learning methods. Finally, we discuss real applications. Noise is one of the main issues when we apply theoretical aspects to solve real-world problems. take place in a clean (noise free) environment and noise often corrupts it. Therefore, we included the study and handling of noise. Then, we discussed multiple application areas. A very important such area is speech processing.

The book is written for students, scientists, researchers, engineers, and practitioners. The readers should have an understanding of basic computational methods as they are taught in the first three semesters of computer science undergraduate curricula. For usage as a textbook, lectures are supported by a systematic presenta-

tion of the material with examples and exercises including background information accompanied with literature discussions.

This work was produced by the authors and was put together in a coherent form. We did our best to give credit to the materials owned by others. We have a reference section at the end of each chapter. In case of any unintentional omission of any work or any part of the text owned by others, we will remain deeply apologetic for this. We read, used, and followed many materials from earlier works done by many other contributors in signal processing and machine learning and generated our work uniquely. We acknowledge these learning and knowledge sharing and remain thankful to all these many contributors.

We would like to thank Ronan Nugent for all supportive arrangements, Bijoy Paul and Michael Sintek for the help from time to time as needed. We also would like to thank the involved Springer team, Celine Chang, Paul Drougas, and Shina Harshavardhan, during this critical transition period for their understanding and cooperation as needed to complete the task.

Kaiserslautern, Germany
Kaiserslautern, Germany
Melbourne, FL, USA
Melbourne, FL, USA
December, 2021

Michael M. Richter
Sheuli Paul
Veton Képuska
Marius Silaghi

Problems, Challenges, and Principal Methods

A very first question is what a signal is. There are different interpretations of this term. The definitions are mainly motivated by the area in which one is interested. Examples are

- Digital signals, such as Boolean values, integers, or real numbers
- Arbitrary objects like words, documents, messages, and images, etc

In this book, the focus is on digital signals. They have no parts and cannot be decomposed, they are simply units. However, several signals, when viewed together, give rise to a signal process. Signals usually do not occur isolated but rather organized in processes. Mostly, such processes are ruled by probabilities, i.e., they are stochastic processes. These occur in two ways: with fixed probabilities (e.g., Markov processes) and with unknown probabilities (Hidden stochastic processes, HSM). We concentrate mainly on the latter ones because this is what we get from the applications. Signal processes contain information. However, this information is not directly visible and is not directly understandable to humans. The information is hidden in properties of the processes. Properties of signal processes are in the initial phase mostly unknown, and we have to find them out, at least approximately. For this, one does not only consider signal processes as they are given, one has rather to manipulate the processes using computations. It is a major challenge to do this properly and depends very much on the type of signals and the application domain.

We view our methods as organized in a machine learning process. To perform this task, one has to deploy many examples of signals. The deployed signals undergo several transformations as necessary. A major difficulty is to find out which transformation one should apply in which situation.

From the hardware point of view, a digital signal processor is a specialized microprocessor with its architecture optimized for the operational needs of digital signal processing. However, this is not the main topic of the book.

Machine learning is a wide area. In this book, we discuss aspects of it related to signal processes. These are mainly unsupervised learning methods.

The properties of signals lead to abstractions. On a more abstract level, one can define properties interesting to applications. In evolution, living organisms have

developed such levels and abstractions over time. Of course, one cannot repeat the evolution. However, it can still be studied because many of the primitive organisms are still alive today. Here, we do not consider evolution, but we can still make use of the observations and results.

The problem is that even a perfect solution to a sub problem is not very useful if it does not help with solutions to other sub problems. The reason is that the sub problems are not contained in modules that can be treated in isolation. Each solution of a sub problem is affected to a degree by the system, or as a whole part.

In a book, one faces therefore the following difficulty:

- How to formulate the overall perspective?
- How to combine solutions to the sub problem in the view of the overall perspective?

The readers should have understanding of basic computational methods as they are taught in the first three semesters of computer science.

Some History The origins of signal processing go back to previous centuries, around the seventeenth century. At that time, it was considered as a part of mathematics, in particular, numerical analysis. In modern times (after 1970) several digital processors have been developed, like in Lincoln Laboratories. In the 1980s, the first chips were developed as Intel 2920, TMS32010 from Texas Instrument, and NEC μ PD7720. In each chapter, there is a section on background information. It contains mainly historical remarks and references for further reading. Here, the discussed choice was the “Dynamic Automatic Noisy Speech Recognition System” (DANSR). The system was devoted to tackle the problem of noises of different kinds that occur simultaneously. This is a major task in the dissertation of the author Sheuli Paul.

A Guide to Different Chapters In each chapter, one finds illustrations of different kinds. The illustrations typically are figures. They may be derived from the peak period when the machine was functional, and also when the machine is working at standard level. Sometimes, illustration is not understood by itself. Therefore, we added some explanations in the figure, for instance in terms of colors. There will be many such figures in the book. For most illustrations, MatLab was used.

Applications are an important part of the book. They are given in the second half of the book. Part C contains the description of several real situations where stochastic processes play a role. In real situations, there is always noise disturbing the stochastic process. The noise can be of a different level. Based on the noise level, we termed the noise as mild, strong, and steady-unsteady. Therefore, we see the probability density function (pdf) of such noisy commands, classified in the legend as mild, unsteady, and strong.

The structure of all chapters is essentially the same. They are structured as

- Overview
- Main text
- Background information

- Exercises
- Reference

We comment shortly on the just-mentioned structure. The overview contains indications of what the reader should study for better understanding before reading the chapter. In addition, this allows finding topics of interest quite fast and it helps a lecturer to select topics for a discussion in class.

The main text contains the contribution of the chapter. In the background, we give the origins to the discussed topics as well as suggestion for further reading. Here, we also present the links to the references. In the main text, there are no references. Please observe that each chapter contains its own list of references. This means that references are listed at the end of each chapter, for the related work discussed in the corresponding chapter text.

The exercises do not contain solutions so that a lecturer can use them in class. In a short way, we now comment on each chapter. This is intended for the readers who want to see what is close to their special interests.

Part I: Realms of Signal Processing

The first part consists of seven chapters, and provides a survey of mathematical and engineering foundations assumed to be known by the reader.

Chapter 1: Digital Signal Representation The details of digital representations of discrete-time signals are presented in this chapter. This includes bridging the gap between the abstract discrete-time signal notation presented in the book, $x[n]$, and its representation in a digital processor.

Chapter 2: Signal Processing Background This chapter introduces briefly the signal processing discipline. It contains the problems and the principal solution methods. In addition, some history is given as well as a short description for the intended readers.

Chapter 3: Fundamentals of Signal Transformation This chapter gives an overview of processing and transforming signals. Here, some basic signal processing formulations are introduced. It is said how the signal processing is an integral part of machine learning and how these two areas are related and how the related outcome of these two areas generates a successful application. In this chapter, we deal with the decomposition of a signal and techniques used. The signal is decomposed into some basic functions and the coefficients. Furthermore, the coefficients then can be used as linear combinations of the signals representing the signal. Transformations such as the z-transform, Fourier transform, discrete cosine transform, local trigonometric transform, and wavelet transform are briefly discussed.

Chapter 4: Digital Filters Digital filters are some mathematical operations to manipulate or analyze the signals. The filter coefficients are generally not variable like in adaptive filters. We introduce the chapter with fundamentals of finite impulse response (FIR) filter and infinite impulse response (IIR) filter, and discuss the applications of some advanced filters.

Chapter 5: Estimation and Detection This chapter discusses some commonly used filters, methods, and techniques to analyze the signal adaptively. The filters use optimization approach in order to analyze the signals. In this chapter, the optimization techniques with examples are discussed. Such techniques are minimum variance unbiased estimation (MVU), best linear unbiased estimator (BLUE), maximum likelihood estimator (MLE), least squares estimator (LSE), minimum least squares estimator (LMSE), and the Bayesian approach.

Chapter 6: Adaptive Signal Processing This chapter has general information about signal modeling using parametric and non-parametric types. It introduces adaptive filters mainly Wiener filter, Kalman filter, and Particle filter.

Chapter 7: Spectral Analysis This chapter considers the problem of the spectral content of the signals by means of parametric and non-parametric signal modeling. In a parametric signal modeling, a signal is analyzed using only few parameters. A non-parametric method has no such parameters quantification. Techniques such as linear prediction analysis using auto-regressive (AR) signal modeling, moving average signal modeling (MA), and auto regressive moving average (ARMA) signal modeling are introduced. Non-parametric signal modeling such as subspace signal analysis applying sub-band coding, discrete cosine transform, and discrete Fourier transform for spectral signal analysis are outlined.

Part II: Machine Learning and Recognition

The ideas of learning with a particular insight into the influence of the learning goals are discussed in eight chapters in this part.

Chapter 8: General Learning This chapter includes the general aspects and principles of learning. An important aspect is the improvement of the learner after learning. In this part, we emphasize on aspects related to stochastic processes as applied to learning.

Chapter 9: Signal Processes, Learning, and Recognition Learning is a very wide and complex area. We cannot give a complete introduction into the subject. Therefore, we restrict ourselves to aspects and methods related to signal processes and signal recognition. The learning considers in particular hidden stochastic

models. Besides a general introduction into the topic, the main topic is Bayesian Learning. Fundamentals of machine learning and its role for signal processing are briefed here.

Chapter 10: Stochastic Processes Here, we introduce the readers to signal processes, in particular to stochastic signal processes. These are partially ordered sets of signals. In the stochastic situation, there is no deterministic rule governing the process. This is rather of statistical character. Here, the main topic is hidden stochastic model. Their main characteristic is that the involved probabilities are unknown.

Chapter 11: Feature Extraction Features are short vectors describing processes. They have to be extracted. This requires techniques described in this chapter. Basic techniques include spectral shaping, windowing the signals, pre-emphasis, folding operators, unfolding, and spectral analysis and its envelope extraction. These techniques are again used in discussed methods such as linear prediction, search, mean squared error minimization, the Baum–Welch algorithm, and the Viterbi algorithm.

Chapter 12: Unsupervised Learning Unsupervised learning techniques such as clustering are introduced here. They are related to learning methods as applied to signal processes. Included in this discussion are K-Means, VQ, EM, and GMM.

Chapter 13: Markov Model and Hidden Stochastic Model (HSM) The topics such as Gaussian mixture models (GMM), the autocorrelation, the Yule–Walker, the co-variance, and the unconstrained least squared (ULS) approach are discussed in this chapter.

Chapter 14: Fuzzy Logic and Rough Sets When dealing with signals, uncertainty is unavoidable. One general method is to use fuzzy sets. In order to obtain decisions that are mostly of binary character, rough sets are introduced. We also discuss fuzzy clustering and fuzzy probabilities.

Chapter 15: Neural Networks This chapter introduces neural networks. In addition to the basics, it includes feed forward networks. Simulation of how the human brain processes the information (biology) is presented briefly, as well as the use of neural networks in machine learning (computers).

Part III: Real Aspects and Applications

Here, the reader is informed about many topics that occur in applications. It contains the required extensions or modifications of techniques that have been introduced in earlier parts.

Chapter 16: Noisy Signals Noise can corrupt the message contained in a signal to a large extent. We discuss different kinds and properties of noise. This is strongly connected to some measurements criteria. Signal-to-noise relation is a basic noise property. This is measured by the average relation of the noise signals to the original signal. There are different kind of noises that require individually tailored techniques.

Chapter 17: Reasoning Methods and Noise Removal Often, an aim is a noise-free signal, thus a main task here is to minimize or remove the noises. An important part of this is noise removal. Noise removal depends very much on noise types. We restrict ourselves to the noise types introduced so far. Each type requires different removal techniques. A major problem occurs when different types of noises occur simultaneously, which requires appropriate removal techniques.

Chapter 18: Audio Signals and Speech Recognition The handling of audio signals has motivated many progresses in signal processing. Speech processes have specific aspects. These are introduced in some details in this chapter.

Chapter 19: Noisy Speech Speech is a major example where noise plays a role. Noise disturbs the reception of speech independent of the type of receiver, which can be a human being or a machine. If a machine is the receiver, some response is expected.

Chapter 20: Aspects of Human Hearing A main characteristic of hearing is its temporal character. This is quite different from dealing with images. The topic is studied because one can obtain elements of recognition of machines from looking at the human ear.

Chapter 21: Speech Features The speech features contain a short and compact representation of the speech. The main condition is that no important information is lost. The term “important” differs from situation to situation and therefore there are several feature extraction algorithms.

Chapter 22: Hidden Stochastic Model for Speech Speech is a typical example for hidden stochastic models. If a person speaks the same words twice, the uttered signals are not the same. The probabilities over the speech elements are unknown and they need to be determined. This chapter delves into mathematical techniques used to model those processes.

Chapter 23: Different Speech Applications—Part A Here, one finds some typical applications. The applications have in common that the speech receiver is a machine. This machine is supposed to execute a command that is contained in the speech. Therefore, the machine has to be embedded in another complex structure that is responsible for the execution as discussed here.

Chapter 24: Different Speech Applications—Part B Different aspects of machine learning of a few practical examples such as Wake-Up-Word (WUW) are introduced. In addition, the Speech Analysis and Sound Effects Laboratory (SASE Lab), Wake-Up-Word: Tool Demo, and Elevator Simulator are presented in this chapter. The aim is to give an essence of understanding, namely how humans communicate using spoken language.

Chapter 25: Biomedical Signals: ECG, EEG These signals are generated by the human body and contain hidden information about it. A major task is to extract such information and to derive the information they contain. There two types of signals that are studied:

- ECG signals
- EEG signals

Chapter 26: Seismic Signals The source of seismic signals is an event that occurs inside the Earth. They contain important information; we can predict the future behavior of the Earth using digital signal processing techniques. A major aspect is to use seismic signals for prediction of earthquakes.

Chapter 27: Radar Signals Radar is an electromagnetic system that detects and locates objects using electromagnetic waves. Transmitter, receiver, and target are basic components of a radar. Radar is categorized in different ways based on areas of application, system types, wave-forms, frequency bands, and antenna types. This chapter is an introduction to radar sensing attached with an application.

Chapter 28: Visual Story Telling is a visual perception of the world. An informed decision about the situation can be made through an effective communication following the perception. Understanding the context, selecting the right tool to display, right selection of data visualization by removing redundancy, orientation, and focus are important for data visualization. This chapter provides essence of VST.

Chapter 29: Digital Processes and Multimedia At the low level, one presents multimedia by using signals. Traditionally, they can be one, two, or three dimensional. Images are a specific example of this, where the signal corresponds to pixels. A major task is to transform signal processes into multimedia representations. Examples of multimedia, techniques, and principles can be seen in this chapter.

Chapter 30: Visualizations for Emergency Operation Centre A visual story is told to manage an emergency environment. This chapter is based on actual emergency situations. The importance of visualization in emergency situations is outlined here. How the visualizations and visual analytics are applied in a typical emergency situation management center, such as an emergency operation center (EOC), is briefly introduced in this chapter.

Chapter 31: Intelligent Interactive Communications Multi-modal communication combines multiple data sources for an effective and meaningful interaction. A concept of multi-modal intelligent interactive machine is proposed in this chapter. This applies automation capabilities to communicate interactively applying verbal and non-verbal multimodalities of human-to-human and human-to-machine communication. The modes of communication can be speech, text, gesture, images, graphics, and visualization.

Chapter 32: Comparisons This chapter presents the main distinctions of the presented chapters.

Contents

Part I Realms of Signal Processing

1	Digital Signal Representation	3
1.1	Introduction	3
1.2	Numbers	4
1.2.1	Numbers and Numerals	5
1.2.2	Types of Numbers	7
1.2.3	Positional Number Systems	8
1.3	Sampling and Reconstruction of Signals	9
1.3.1	Scalar Quantization	11
1.3.2	Quantization Noise	15
1.3.3	Signal-to-Noise Ratio	19
1.3.4	Transmission Rate	22
1.3.5	Nonuniform Quantizer	23
1.3.6	Companding	24
1.4	Data Representations	24
1.4.1	Fixed-Point Number Representations	25
1.4.2	Sign-Magnitude Format	26
1.4.3	One's-Complement Format	27
1.4.4	Two's-Complement Format	28
1.5	Fix-Point DSP's	29
1.6	Fixed-Point Representations Based on Radix-Point	31
1.7	Dynamic Range	34
1.8	Precision	34
1.9	Background Information	35
1.10	Exercises	38
	References	38
2	Signal Processing Background	39
2.1	Basic Concepts	39
2.2	Signals and Information	41
2.3	Signal Processing	41

2.4	Discrete Signal Representations	42
2.5	Delta and Impulse Function	43
2.6	Parseval's Theorem	45
2.7	Gibbs Phenomenon	47
2.8	Wold Decomposition	47
2.9	State Space Signal Processing	49
2.10	Common Measurements	52
2.10.1	Convolution	52
2.10.2	Correlation	53
2.10.3	Auto Covariance	54
2.10.4	Coherence	55
2.10.5	Power Spectral Density (PSD)	56
2.10.6	Estimation and Detection	59
2.10.7	Central Limit Theorem	60
2.10.8	Signal Information Processing Types	61
2.10.9	Machine Learning	61
2.10.10	Exercises	65
	References	66
3	Fundamentals of Signal Transformations	69
3.1	Transformation Methods	71
3.1.1	Laplace Transform	72
3.1.2	Z-Transform	73
3.1.3	Fourier Series	76
3.1.4	Fourier Transform	76
3.1.5	Discrete Fourier Transform and Fast Fourier Transform	77
3.1.6	Zero Padding	78
3.1.7	Overlap-Add and Overlap-Save Convolution Algorithms	78
3.1.8	Short Time Fourier Transform (STFT)	79
3.1.9	Wavelet Transform	88
3.1.10	Windowing Signal and the DCT Transforms	91
3.2	Analysis and Comparison of Transformations	92
3.3	Background Information	93
3.4	Exercises	94
	References	95
4	Digital Filters	97
4.1	Introduction	97
4.1.1	FIR and IIR Filters	99
4.1.2	Bilinear Transform	104
4.2	Windowing for Filtering	104
4.3	Allpass Filters	106
4.4	Lattice Filters	107
4.5	All-Zero Lattice Filter	111

4.6	Lattice Ladder Filters	111
4.7	Comb Filter	111
4.8	Notch Filter.....	112
4.9	Background Information.....	114
4.10	Exercises	114
	References.....	115
5	Estimation and Detection	117
5.1	Introduction	117
5.2	Hypothesis Testing	118
5.2.1	Bayesian Hypothesis Testing.....	118
5.2.2	MAP Hypothesis Testing	119
5.3	Maximum Likelihood (ML) Hypothesis Testing	120
5.4	Standard Estimation Techniques	120
5.4.1	Minimum Variance Unbiased (MVU) Estimator.....	122
5.4.2	Best Linear Unbiased Estimator (BLUE).....	123
5.4.3	Maximum Likelihood Estimator (MLE).....	124
5.4.4	Least Squares Estimator (LSE).....	125
5.4.5	Linear Minimum Mean Square Error Estimator (LMMSE).....	128
5.5	Exercises	130
	References.....	130
6	Adaptive Signal Processing	131
6.1	Introduction	131
6.2	Parametric Signal Modeling.....	132
6.2.1	Parametric Estimation	132
6.3	Wiener Filtering.....	132
6.4	Kalman Filter.....	135
6.4.1	Smoothing.....	136
6.5	Particle Filter	137
6.6	Fundamentals of Monte Carlo	139
6.6.1	Importance Sampling (IS)	140
6.7	Non-Parametric Signal Modeling	143
6.8	Non-Parametric Estimation.....	143
6.8.1	Correlogram	144
6.8.2	Periodogram	144
6.9	Filter Bank Method	145
6.10	Quadrature Mirror Filter Bank (QMF)	148
6.11	Background Information.....	149
6.12	Exercises	150
	References.....	150
7	Spectral Analysis	151
7.1	Introduction	151
7.2	Adaptive Spectral Analysis.....	152

7.3	Multivariate Signal Processing	152
7.3.1	Sub-Band Coding and Subspace Analysis	155
7.4	Wavelet Analysis	156
7.5	Adaptive Beam Forming	159
7.6	Independent Component Analysis (ICA)	162
7.7	Principal Component Analysis (PCA)	164
7.8	Best Basis Algorithms	166
7.9	Background Information	166
7.10	Exercises	167
	References	167
Part II Machine Learning and Recognition		
8	General Learning	173
8.1	Introduction to Learning	173
8.2	The Learning Phases	175
8.2.1	Search and Utility	175
8.3	Search	176
8.3.1	General Search Model	176
8.3.2	Preference Relations	177
8.3.3	Different Learning Methods	178
8.3.4	Similarities	179
8.3.5	Learning to Recognize	179
8.3.6	Learning Again	180
8.4	Background Information	181
8.5	Exercises	182
	References	182
9	Signal Processes, Learning, and Recognition	183
9.1	Learning	183
9.2	Bayesian Formalism	186
9.2.1	Dynamic Bayesian Theory	186
9.2.2	Recognition and Search	186
9.2.3	Influences	188
9.3	Subjectivity	189
9.4	Background Information	189
9.5	Exercises	191
	References	191
10	Stochastic Processes	193
10.1	Preliminaries on Probabilities	194
10.2	Basic Concepts of Stochastic Processes	194
10.2.1	Markov Processes	195
10.2.2	Hidden Stochastic Models (HSM)	196
10.2.3	HSM Topology	198
10.2.4	Learning Probabilities	198
10.2.5	Re-estimation	199

10.2.6	Redundancy	199
10.2.7	Data Preparation	200
10.2.8	Proper Redundancy Removal	200
10.3	Envelope Detection	201
10.3.1	Silence Threshold Selection	203
10.3.2	Pre-emphasis	203
10.4	Several Processes	206
10.4.1	Similarity	207
10.4.2	The Local-Global Principle	209
10.4.3	HSM Similarities	214
10.5	Conflict and Support	215
10.6	Examples and Applications	215
10.7	Predictions	216
10.8	Background Information	217
10.9	Exercises	218
	References	219
11	Feature Extraction	221
11.1	Feature Extractions	222
11.2	Basic Techniques	223
11.2.1	Spectral Shaping	223
11.3	Spectral Analysis and Feature Transformation	229
11.3.1	Parametric Feature Transformations and Cepstrum	229
11.3.2	Standard Feature Extraction Techniques	230
11.3.3	Frame Energy	230
11.4	Linear Prediction Coefficients (LPC)	231
11.5	Linear Prediction Cepstral Coefficients (LPCC)	234
11.6	Adaptive Perceptual Local Trigonometric Transformation (APLTT)	235
11.7	Search	235
11.7.1	General Search Model	235
11.8	Predictions	237
11.8.1	Purpose	237
11.8.2	Linear Prediction	237
11.8.3	Mean Squared Error Minimization	239
11.8.4	Computation of Probability of an Observation Sequence	240
11.8.5	Forward and Backward Prediction	242
11.8.6	Forward-Backward Prediction	244
11.9	Background Information	248
11.10	Exercises	249
	References	250
12	Unsupervised Learning	251
12.1	Generalities	251
12.2	Clustering Principles	252

12.3	Cluster Analysis Methods	253
12.4	Special Methods	254
12.4.1	K-Means.....	255
12.4.2	Vector Quantization (VQ)	256
12.4.3	Expectation Maximization (EM).....	257
12.4.4	GMM Clustering.....	258
12.5	Background Information.....	259
12.6	Exercises	260
	References	260
13	Markov Model and Hidden Stochastic Model	261
13.1	Markov Process	261
13.2	Gaussian Mixture Model (GMM)	262
13.3	Advantages of Using GMM	263
13.4	Linear Prediction Analysis	263
13.4.1	Autocorrelation Method	267
13.4.2	Yule-Walker Approach	268
13.4.3	Covariance Method	269
13.4.4	Comparison of Correlation and Covariance Methods ..	270
13.5	The ULS Approach	271
13.6	Comparison of ULS and Covariance Methods	273
13.7	Forward Prediction.....	274
13.8	Backward Prediction	275
13.9	Forward-Backward Prediction	276
13.10	Baum-Welch Algorithm	276
13.11	Viterbi Algorithm	277
13.12	Background Information.....	277
13.13	Exercises	278
	References	278
14	Fuzzy Logic and Rough Sets	281
14.1	Rough Sets	281
14.2	Fuzzy Sets	283
14.2.1	Basis Elements	283
14.2.2	Possibility and Necessity	285
14.3	Fuzzy Clustering	286
14.4	Fuzzy Probabilities	287
14.5	Background Information.....	288
14.6	Exercises	288
	References	289
15	Neural Networks	291
15.1	Neural Network Types	292
15.1.1	Neural Network Training	293
15.1.2	Neural Network Topology	294

15.2	Parallel Distributed Processing	297
15.2.1	Forward and Backward Uses	298
15.2.2	Learning	299
15.3	Applications to Signal Processing	299
15.4	Background Information	300
15.5	Exercises	301
	References	301

Part III Real Aspects and Applications

16	Noisy Signals	307
16.1	Introduction	307
16.2	Noise Questions	311
16.3	Sources of Noise	311
16.4	Noise Measurement	311
16.5	Weights and A-Weights	312
16.6	Signal to Noise Ratio (SNR)	315
16.7	Noise Measuring Filters and Evaluation	317
16.8	Types of Noise	317
16.9	Origin of Noises	320
16.10	Box Plot Evaluation	320
16.11	Individual Noise Types	321
16.11.1	Residual	321
16.11.2	Mild	321
16.11.3	Steady-Unsteady Time Varying Noise	322
16.11.4	Strong Noise	323
16.12	Solution to Strong Noise: Matched Filter	324
16.13	Background Information	325
16.14	Exercises	326
	References	326
17	Reasoning Methods and Noise Removal	327
17.1	Generalities	327
17.2	Special Noise Removal Methods	328
17.2.1	Residual Noise	328
17.2.2	Mild Noise	328
17.2.3	Steady-Unsteady Noise	330
17.2.4	Strong Noise	330
17.3	Poisson Distribution	331
17.3.1	Outliers and Shots	331
17.3.2	Underlying Probability of Shots	332
17.4	Kalman Filter	332
17.4.1	Prediction Estimates	335
17.4.2	White Noise Kalman Filtering	335
17.4.3	Application of Kalman Filter	336

17.5	Classification, Recognition and Learning.....	337
17.5.1	Summary of the Used Concepts.....	338
17.6	Principle Component Analysis (PCA)	338
17.7	Reasoning Methods	340
17.7.1	Case-Based Reasoning (CBR)	341
17.8	Background Information.....	341
17.9	Exercises	342
	References.....	342
18	Audio Signals and Speech Recognition.....	345
18.1	Generalities of Speech	345
18.2	Categories of Speech Recognition	347
18.3	Automatic Speech Recognition	349
18.3.1	System Structure	350
18.4	Speech Production Model	350
18.5	Acoustics	351
18.6	Human Speech Production	351
18.6.1	The Human Speech Generation	352
18.6.2	Excitation.....	353
18.6.3	Voiced Speech.....	355
18.6.4	Unvoiced Speech.....	357
18.7	Silence Regions	359
18.8	Glottis	360
18.9	Lips	360
18.10	Plosive Speech Source	361
18.11	Vocal-Tract	361
18.12	Parametric and Non-parametric Models	363
18.13	Formants.....	366
18.14	Strong Noise	366
18.15	Background Information.....	367
18.16	Exercises	367
	References.....	367
19	Noisy Speech	369
19.1	Introduction	369
19.2	Colored Noise	369
19.2.1	Additional Types of Colored Noise	371
19.3	Poisson Processes and Shots	372
19.4	Matched Filters.....	374
19.5	Shot Noise	375
19.6	Background Information.....	377
19.7	Exercises	378
	References.....	378
20	Aspects of Human Hearing	379
20.1	Human Ear	380
20.2	Human Auditory System	380

20.3	Critical Bands and Scales	382
20.3.1	Mel Scale	382
20.3.2	Bark Scale	383
20.3.3	Erb Scale	384
20.3.4	Greenwood Scale	384
20.4	Filter Banks	385
20.4.1	ICA Network	386
20.4.2	Auditory Filter Banks	387
20.4.3	Filter Banks	387
20.4.4	Mel Critical Filter Bank	388
20.5	Psycho-Acoustic Phenomena	389
20.5.1	Perceptual Measurement	389
20.5.2	Human Hearing and Perception	389
20.5.3	Sound Pressure Level (SPL)	390
20.5.4	Absolute Threshold of Hearing (ATH)	391
20.6	Perceptual Adaptation	392
20.7	Auditory System and Hearing Model	392
20.8	Auditory Masking and Masking Frequency	393
20.9	Perceptual Spectral Features	394
20.10	Critical Band Analysis	394
20.11	Equal Loudness Pre-emphasis	395
20.12	Perceptual Transformation	396
20.13	Feature Transformation	396
20.14	Filters and Human Ear	396
20.15	Temporal Aspects	397
20.16	Background Information	398
20.17	Exercises	399
	References	399
21	Speech Features	401
21.1	Generalities	401
21.2	Cost Functions	402
21.3	Special Feature Extractions	403
21.3.1	MFCC Features	403
21.3.2	Feature Transformation Applying DCT	407
21.4	Background Information	412
21.5	Exercises	413
	References	413
22	Hidden Stochastic Model for Speech	415
22.1	Generals	415
22.2	Hidden Stochastic Model	417
22.3	Forward and Backward Predictions	421
22.3.1	Forward Algorithm	421
22.3.2	Backward Algorithm	422
22.4	Forward-Backward Prediction	423

22.5	Burg Approach	424
22.6	Graph Search	426
22.6.1	Recognition Model with Search	427
22.7	Semantic Issues and Industrial Applications	428
22.8	Problems with Noise	430
22.9	Aspects of Music	430
22.10	Music Reception	433
22.11	Background Information	433
22.12	Exercises	434
	References	434
23	Different Speech Applications—Part A	437
23.1	Generalities	437
23.2	Example Applications	437
23.2.1	Experimental Laboratory	439
23.2.2	Health Care Support (Everyday Actions)	441
23.2.3	Diagnostic Support for Persons with Possible Dementia	444
23.2.4	Noise	446
23.3	Background Information	447
23.4	Exercises	449
	References	449
24	Different Speech Applications—Part B	451
24.1	Introduction	451
24.2	Discrete-Time Signals	454
24.3	Speech Processing	456
24.3.1	Framing	457
24.3.2	Pre-emphasis	458
24.3.3	Windowing	460
24.3.4	Fourier Transform	462
24.3.5	Mel-Filtering	465
24.3.6	Mel-Frequency Cepstral Coefficients	467
24.4	Speech Analysis and Sound Effects Laboratory (SASE_Lab)	470
24.5	Wake-Up-Word Speech Recognition	475
24.5.1	Introduction	476
24.5.2	Wake-Up-Word Paradigm	476
24.5.3	Wake-Up-Word: Definition	478
24.5.4	Wake-Up-Word System	479
24.5.5	Front-End of the Wake-Up-Word System	480
24.6	Conclusion	492
24.6.1	Wake-Up-Word: Tool Demo	493
24.6.2	Elevator Simulator	494
24.7	Background Information	494
24.8	Exercises	496
24.9	Speech Analysis and Sound Effects Laboratory (SASE_Lab)"	497
	References	497

25	Biomedical Signals: ECG, EEG	499
25.1	ECG Signals	500
25.1.1	Bioelectric Signals	500
25.1.2	Noise	502
25.2	EEG Signals	503
25.2.1	General Properties	503
25.2.2	Signal Types and Properties	504
25.2.3	Disadvantages	504
25.3	Neural Network Use	505
25.4	Major Research Questions	506
25.5	Background Information	506
25.6	Exercises	507
	References	508
26	Seismic Signal	509
26.1	Generalities	510
26.2	Sources of Seismic Signals	510
26.3	Intermediate Elements	511
26.4	Practical Data Sources	511
26.5	Major Seismic Problems	512
26.6	Noise	513
26.7	Background Information	514
26.8	Exercises	514
	References	515
27	Radar Signals	517
27.1	Introduction	517
27.2	Radar Types and Applications	519
27.3	Doppler Equations, Ambiguity Function (AF) and Matched Filter	520
27.4	Moving Target Detection	521
27.5	Applications and Discussions	523
27.6	Examples	523
27.7	Background Information	524
27.8	Exercises	526
	References	529
28	Visual Story Telling	531
28.1	Introduction	531
28.1.1	Common Visualization Approaches	533
28.2	Analytics and Visualization	533
28.2.1	Exploratory Data Analysis	534
28.2.2	Visual Data Mining	535
28.3	Communication and Visualization	535
28.4	Background Information	536

28.5 Exercises	539
References	539
29 Digital Processes and Multimedia	541
29.1 Images	541
29.1.1 Digital Image Processing	542
29.1.2 Images as Matrices	549
29.1.3 Gray Scale Images	550
29.2 Spatial Filtering	552
29.2.1 Linear Filtering of Images	552
29.2.2 Separable Filters	553
29.2.3 Mechanics of Linear Spatial Filtering Operation	555
29.3 Median Filtering	556
29.4 Color Equalization	557
29.4.1 Image Transformations	559
29.4.2 Examples of Image Transformation Matrixes	564
29.5 Basic Image Statistics	567
29.6 Abstraction Levels of Images and Its Representations	569
29.6.1 Lowest Level	570
29.6.2 Geometric Level	571
29.6.3 Domain Level	571
29.6.4 Segmentation	572
29.7 Background Information	572
29.8 Exercises	573
References	574
30 Visualizations of Emergency Operation Centre	575
30.1 Introduction	575
30.2 Communications in Emergency Situations	576
30.3 Emergency Scenario	577
30.3.1 Classification and EOC Scenario	577
30.4 Technical Aspects and Techniques	578
30.4.1 Classification	578
30.4.2 Clustering	579
30.5 Background Information	579
30.6 Exercises	580
References	581
31 Intelligent Interactive Communications	583
31.1 Introduction	583
31.2 Spoken Dialogue System	585
31.3 Gesture Based Interaction	588
31.4 Object Recognition and Identification	588
31.5 Visual Story Telling	589
31.6 Virtual Environment for Personal Assistance	591
31.7 Sensor Fusion	592

31.8	Intelligent Human Machine for Communication: Application Scenario.....	593
31.9	Background Information.....	593
31.10	Exercises	595
	References.....	595
32	Comparisons	597
32.1	Generalities.....	597
32.1.1	EEG and ECG.....	600
32.1.2	Speech and Biomedical Applications.....	601
32.1.3	Seismic and Biomedical Signals	601
32.1.4	Speech and Images.....	601
32.2	Overall.....	602
32.3	Background Information.....	603
32.3.1	General	603
32.4	Exercises	603
	Glossary	605

List of Figures

Fig. 1.1	The interfacing with a continuous signal that is optionally conditioned by the sensor conditioner. Analog-to-Digital Conversion: (a) Continuous signal $x(t)$, (b) Sampled signal $x_a(t)$ with sampling period T satisfying Nyquist rate as specified by Sampling Theorem, (c) Digital sequence $x[n]$ obtained after sampling and quantization	10
Fig. 1.2	Digital-to-analog conversion. (a) Processed digital signal $y[n]$. (b) Continuous signal representation $y_a(nT)$. (c) Low-pass filtered continuous signal $y(t)$	11
Fig. 1.3	Conceptual representation of ADC	12
Fig. 1.4	Example of a uniform quantization for $L = 16$ levels. As it is discussed in the following sections, $L = 16$ levels, require 4 bits of the codeword to represent each level. Because the distribution of the input values is uniform the decision and reconstruction levels are uniformly spaced: $\hat{x} = Q(x)$	13
Fig. 1.5	Example of pdf's (probability distribution function)	15
Fig. 1.6	Subplots are enumerated from top to bottom	17
Fig. 1.7	Example of slowly varying signal that causes quantization error to be correlated	18
Fig. 1.8	Example of speech signal demonstrating the effect of step size to the degree of correlation of quantization error	19
Fig. 1.9	Histogram of quantization error for speech signal. Note the reduction of error magnitude as well as increase of uniformity of the distribution with increase of number of quantization levels	20
Fig. 1.10	Histogram of quantization error for speech signal after adding Gaussian noise with zero mean and variance of 5	21
Fig. 1.11	Uniform probability distribution function $p(e)$ of error signal in the range $[-\Delta/2, \Delta/2]$	22

Fig. 1.12	Block diagram of companding in the transmitting and receiving <i>DSP</i> system. The $x[n]$ is an unquantized input sample with a nonuniform <i>pdf</i> of values; $y[n]$ is the value obtained after nonlinear transformation with uniform <i>pdf</i> values, and $\tilde{y}[n]$ is quantized sample, $c[n]$ is encoded binary representation of this sample value. This binary encoded stream is typically transmitted to a receiving system where it is converted back to the original by decoding the input encoded samples $c'[n]$ to $y'[n]$, and applying the inverse of the non-linear transformation T^{-1} obtaining sequence $x'[n]$. If $c'[n] = c[n]$, then $x'[n]$ differs from $x[n]$ by amount of introduced quantization noise	25
Fig. 1.13	Signed magnitude format	26
Fig. 1.14	The 16-bit unsigned and signed data type's representation	31
Fig. 1.15	Possible $\mathbb{Q}_{p,q}$ format representations of 16-bit data length	36
Fig. 2.1	Signal is processed from continuous to discrete representation	42
Fig. 2.2	Delta function	43
Fig. 2.3	Rectangular waveform and Gibbs phenomenon applying successive approximations 40 sine waveforms	48
Fig. 2.4	Rectangular waveform and Gibbs' phenomenon approximating $n = 1, 3, 7, 19, 49$ and 70 sine wave-forms	49
Fig. 2.5	Continuous and discrete SISO system	50
Fig. 2.6	SISO and MIMO model analysis	50
Fig. 2.7	The state space representation of a system	52
Fig. 2.8	Input, system and output signals	56
Fig. 2.9	Standard error	61
Fig. 2.10	Time range	61
Fig. 2.11	Exploring problems in machine learning	64
Fig. 3.1	The 64 point Hanning window	80
Fig. 3.2	The 64 point Hamming window	81
Fig. 3.3	The 64 point Kaizer window with parameter $\alpha = 7$	81
Fig. 3.4	The 64 point Chebyshev window with attenuation parameter 100.....	82
Fig. 3.5	Rising cut-off function	92
Fig. 3.6	Trapezoidal signal	94
Fig. 4.1	The figure shows an ideal low-pass filter specification. The F_{pass} depicts pass-band, F_s depicts the sampling period, while the other notation is self-explanatory	98
Fig. 4.2	The figure depicts the steps involved in periodic discrete-time sequence spectrum analysis using DFT	104

Fig. 4.3	A basic structure of one state allpass filter	107
Fig. 4.4	A two stage allpass filter	107
Fig. 4.5	A basic structure of allpass filter.....	108
Fig. 4.6	Transformation of a lattice filter and its transfer function from A(z) to D(z) and D(z) to A(z).....	108
Fig. 4.7	M stage allpass lattice filter	108
Fig. 4.8	An M stage lattice filter	109
Fig. 4.9	Comb filter magnitude and frequency response.....	112
Fig. 4.10	Notch filter and its frequency response.....	113
Fig. 5.1	Standard adaptive signal processing approach	118
Fig. 6.1	Stochastic signal processing	132
Fig. 6.2	A basic structure of two channel filter bank	145
Fig. 7.1	Multivariate signal processing	153
Fig. 7.2	Decimation and Interpolation	153
Fig. 7.3	Sub-band coding	155
Fig. 7.4	Basic Structure of a filter bank	160
Fig. 7.5	ICA separates the mixtures into its three different sources	165
Fig. 8.1	Learning	174
Fig. 8.2	The learning phases	175
Fig. 8.3	Simulated stock value	180
Fig. 9.1	Typical machine learning process	184
Fig. 9.2	General supervised learning	184
Fig. 9.3	Bayes rule for computing probabilities	187
Fig. 10.1	The figure shows 1st order magnitude response of low-pass and high-pass filters. (a) <i>High-Pass Filter</i> . (b) <i>Low-Pass Filter</i>	195
Fig. 10.2	Example of a Markov chain	196
Fig. 10.3	Envelop detection by LPF method and Hilbert transform	202
Fig. 10.4	The (a) original frequency representation and its (b) Hilbert Transform representation	203
Fig. 10.5	Example of redundancy removal	204
Fig. 10.6	Block diagram of a pre-emphasis Filter with a_{em} factor	205
Fig. 10.7	Pre-emphasis filter characteristics given by amplitude and phase response	205
Fig. 10.8	DTW frame depicting allowable paths	213
Fig. 11.1	Main processing steps before feature extraction and prediction	222
Fig. 11.2	Pre-filtering and pre-emphasized signal. (a) Redundancy removal speech. (b) Prefiltered redundancy removal speech. (c) Speech spectrum. (d) Prefiltered emphasized spectrum	224
Fig. 11.3	Frame of Speech with corresponding Spectrum, and its pre-emphasized versions	225
Fig. 11.4	Spectral shaping and analysis	226
Fig. 11.5	Windowing and folding in wavelet packet analysis	228

Fig. 11.6	Block diagram of procedure deployed computing cepstrum	229
Fig. 11.7	Example of 256 tap (points) Hanning window	232
Fig. 11.8	Example of 256 tap (points) Hamming window	233
Fig. 11.9	Block diagram depicting computation of LPC features	233
Fig. 11.10	LPCC feature extraction	234
Fig. 11.11	Viterbi decoding in speech pattern recognition	236
Fig. 11.12	Linear prediction analysis	238
Fig. 11.13	Viterbi Algorithm as applied to the network of the type shown	246
Fig. 11.14	Viterbi Algorithm as applied to text	247
Fig. 12.1	Unsupervised learning	252
Fig. 12.2	Results of application of K-means procedure	255
Fig. 12.3	Vector quantization	256
Fig. 13.1	Gaussian Mixture example	262
Fig. 13.2	Short-time prediction widow	265
Fig. 13.3	Covariance method applied to a frame of speech	271
Fig. 13.4	ULS based Prediction Coefficients and pole-zero locations of the filter and inverse filter	272
Fig. 13.5	Residual of the inverse ULS filter	273
Fig. 13.6	Frequency response of ULS and covariance approach	273
Fig. 14.1	Decisions using rough sets	282
Fig. 14.2	Visualisation of rough sets	282
Fig. 14.3	Fuzzy set with one dimensional data	286
Fig. 14.4	Representation of strict membership	287
Fig. 14.5	Sigmoid membership function	287
Fig. 15.1	An example of artificial neural network	292
Fig. 15.2	An example of an artificial neuron	296
Fig. 15.3	Example of activation function utilized by a neuron	297
Fig. 15.4	An example of feed-forward neural network	299
Fig. 16.1	Example of the waveform (e.g. speech Data), pitch (e.g., pitch data), and spectrographic display (e.g. spectrogram) of clean spoken utterance ‘oeffne die tuer’ in a standard/quiet environment	308
Fig. 16.2	Example of the waveform (e.g. speech data), pitch (e.g., pitch data), and spectrographic display (e.g. spectrogram) of speech ‘oeffne die tuer’ recorded in an noisy industrial environment. Various types of noises were present due to heavy duty of machinery and running motors providing noisy environment. These machines were in resting state	308

Fig. 16.3	Example of the waveform (e.g. speech data), pitch (e.g., pitch data), and spectrographic display (e.g. spectrogram) of speech ‘oeffnedietuer’ recorded in an noisy industrial environment. Various types of noises were present due to heavy duty of machinery and running motors providing noisy environment. These machines were in running state generating varying load sounds. Such industrial environment have always varying types of noisy sounds. In such environments, the noise can be so high that speaker can not distinguish the speech from noise	309
Fig. 16.4	Linear microphone array for signal processing in a typical noisy environment. Number of microphones m_i for $i = 1, \dots, 8$ capturing the signal. Arrival times t_i for $i = 1, \dots, 8$ of the signals in the array. Signal is captured at different times due to varying paths of arrivals of the wave pattern	310
Fig. 16.5	Example of the Noisy Waveform and its Spectrographic display	310
Fig. 16.6	Example of the A weighted filter measurement using varying noisy signal	314
Fig. 16.7	Example of the A weighted filter measurement of the strong noisy signal	315
Fig. 16.8	The A weighted filter measurement of different noises	316
Fig. 16.9	Hybrid noise	318
Fig. 16.10	Energy for the different noisy types	319
Fig. 16.11	Different noisy signal energies	319
Fig. 16.12	Boxplot chart of sound level measurement by A-weighting filter	321
Fig. 16.13	Boxplot chart of sound level measurement by A-weighting filter	323
Fig. 16.14	Noisy and enhanced signal spectrum	324
Fig. 16.15	Example of speech signal mixed with a strong noise and its handling	325
Fig. 17.1	Aliased-free down-sampler	328
Fig. 17.2	Flow diagram of the Kalman prediction	334
Fig. 17.3	Steady-unsteady nosy signal enhanced by Kalman filtering. (a) Noisy spoken spectrum: Oeffne die Tuer. (b) Enhanced spoken spectrum applying M-band Kalman filter Oeffne die Tuer	336
Fig. 17.4	Example of applied PCA to noisy signals. (a) Standard enhanced spectrum applying PCA: Oeffne die Tuer. (b) Special enhanced spectrum applying PCA: Oeffne die Tuer	340
Fig. 18.1	The speech cycle from human perspective	346
Fig. 18.2	Speech recognition by a machine	347

Fig. 18.3	General speech recognition system	348
Fig. 18.4	A more detailed speech recognition system	348
Fig. 18.5	Bayes rule in speech recognition system problem	349
Fig. 18.6	The human parts involved in speech production	353
Fig. 18.7	Basic US English sounds classification	354
Fig. 18.8	Delta functions	355
Fig. 18.9	Periodic pulses	356
Fig. 18.10	Source model of periodic pulses	356
Fig. 18.11	Voiced speech in the source excitation model	356
Fig. 18.12	Unvoiced speech in the source excitation model	358
Fig. 18.13	Voiced and unvoiced speech example	359
Fig. 18.14	Plosive sound generation in source excitation speech model	361
Fig. 18.15	Discrete Vocal Tract model of a number of uniform tubes	362
Fig. 18.16	Source excitation speech model	363
Fig. 18.17	Parametric speech source modeling using ARMA model	364
Fig. 18.18	Moving average and regressive processes	365
Fig. 19.1	Gaussian model for colored noise. (a) Colored noise. (b) Autocorrelation of colored noise. (c) Frequency information of colored noise	370
Fig. 19.2	Shaping filter where $y[n] = ay[n - 1] + (1 - a)x[n]$	371
Fig. 19.3	Output of matched filtering	374
Fig. 19.4	Additive noise corrupted signal	375
Fig. 20.1	Figure depicting human ear. https://commons.wikimedia.org/w/index.php?curid=1678037	381
Fig. 20.2	Mel scale	383
Fig. 20.3	Bark scale	384
Fig. 20.4	Erb scale	385
Fig. 20.5	Comparison of different scales	386
Fig. 20.6	Threshold of hearing	392
Fig. 20.7	Filters in the human Ear	397
Fig. 21.1	MFCC feature extraction	404
Fig. 21.2	Example of pre-emphasised data computed from TIMIT corpus: TIMIT/TRAIN/DR1/FSJK1/SI696.WAV	405
Fig. 21.3	Example of FFT data computed from TIMIT corpus: TIMIT/TRAIN/DR1/FSJK1/SI696.WAV	405
Fig. 21.4	Triangles of different bandwidths following Mel scale	406
Fig. 21.5	Block diagram of cepstrum features generation	407
Fig. 21.6	Block diagram of PLP features generation	408
Fig. 21.7	Linear Prediction following the source excitation model	409
Fig. 21.8	Block diagram of LPC feature extraction	410
Fig. 21.9	Block diagram of LPCC feature extraction	410
Fig. 21.10	Block diagram of PLPCC feature extraction	411
Fig. 21.11	Block diagram of SILT feature extraction	412
Fig. 22.1	Signal modeling	420

Fig. 22.2	An hypothetical example of a 3 state HMM	421
Fig. 22.3	Forward-backward prediction	423
Fig. 22.4	The Burg approach	424
Fig. 22.5	Visualization of forward and backward prediction	425
Fig. 22.6	Visualization of hypothetical incorrect classification	426
Fig. 22.7	HSM using hidden Markov modeling (HMM) for acoustic and language search in recognition	427
Fig. 22.8	HSM in DANSR model definition	427
Fig. 22.9	The DANSR system	428
Fig. 22.10	Overall view of ambient assistive living room for a nursing home	430
Fig. 22.11	Musical notation in a score	432
Fig. 23.1	Analysis component of health care support	443
Fig. 23.2	Probability distribution of DTW distance score using 100 training and 25 test data	447
Fig. 24.1	Search and decoding process is depicted in the figure above	452
Fig. 24.2	Figure depicting human anatomy and the organs involved in speech production. http://training.seer.cancer.gov/head-neck/anatomy/overview.html , https://commons.wikimedia.org/w/index.php?curid=1678037	453
Fig. 24.3	Figure depicting block diagram of speech recognition system	453
Fig. 24.4	Continuous and discrete time signal	455
Fig. 24.5	Continuous and discrete time signal with reduced sampling rate	456
Fig. 24.6	Continuous and discrete time signal sampled at the higher rate of 44.1 kHz	456
Fig. 24.7	Front-end stage of a typical Mel-filter Cepstral Coefficients based speech recognition system	457
Fig. 24.8	Speech signal with overlapping frames	458
Fig. 24.9	Pre-emphasis filter block diagram	458
Fig. 24.10	Pre-emphasis filter frequency response	459
Fig. 24.11	Original and pre-emphasised signal	459
Fig. 24.12	Hamming Window of length 256 and its frequency characteristics depicted to the right	461
Fig. 24.13	Figure depicting: (a) Speech signal with overlaid Hamming Window. (b) Speech signal after the windowing operation	461
Fig. 24.14	Spectrum of a speech frame depicting its content in frequency	463
Fig. 24.15	Mel-frequency transformation curve	466
Fig. 24.16	Normalized triangular shaped filters following Mel-scale	467
Fig. 24.17	Linear scale vs Mel scale spectrum	468

Fig. 24.18	Example of a mel-frequency Cepstral Coefficients of order 12 of a frame of speech	469
Fig. 24.19	Image depicting the interfacing provided by the MATLAB tool SASE_Lab “Speech Analysis and Special Effects Laboratory”	471
Fig. 24.20	Waveform and Spectrograph analysis performed on the data	472
Fig. 24.21	3 seconds of data selected from the spectrogram depicted in red-color from the waveform	473
Fig. 24.22	3 seconds of data depicted with corresponding spectrogram computed from the selected section of waveform data	473
Fig. 24.23	Vowels, 'a', 'e', 'i', 'o', 'u' uttered and captured	474
Fig. 24.24	Vowels, 'a', 'i', 'e', 'u'	475
Fig. 24.25	Wake-up-word speech recognition system	480
Fig. 24.26	Speech signal with voice activity detector segmentation, spectrogram, and enhanced spectrogram generated by the front end module (K��puska and Klein 2009)	481
Fig. 24.27	Support vector machine decision surface as computed with linear SVM	482
Fig. 24.28	Scoring of plain WUW recognition	484
Fig. 24.29	Score1 vs. Score2 example (K��puska and Klein 2009)	485
Fig. 24.30	Score1 vs. Score3 example (K��puska and Klein 2009)	486
Fig. 24.31	Score1 vs. Score2 linear discrimination surface (K��puska and Klein 2009)	487
Fig. 24.32	Score1 vs. Score3 linear discrimination surface (K��puska and Klein 2009)	487
Fig. 24.33	Score1 vs. Score2 vs. Score3 hyper-plane discrimination surface (K��puska and Klein 2009)	488
Fig. 24.34	Distribution of Scores for triple-scoring and linear SVM (K��puska and Klein 2009)	488
Fig. 24.35	Scatter plot of INV and OOV distributions of Scores1 vs Score2 with RBF discriminating function (K��puska and Klein 2009)	489
Fig. 24.36	Scatter plot of INV and OOV distributions of Scores1 vs Score3 with RBF discriminating function (K��puska and Klein 2009)	490
Fig. 24.37	Discriminating surface presented in 3D (K��puska and Klein 2009)	491
Fig. 24.38	PDF of INV and OOV distributions for RBF (K��puska and Klein 2009)	491
Fig. 24.39	PDF of INV and OOV distributions for RBF utilizing more data (K��puska and Klein 2009)	492

Fig. 24.40	Example of “no Wake-up-Word”/some other word spoken, and the case when “Wake-up-Word” was spoken not triggering off and triggering the system as indicated with different color as well as audibly played sound “yes”. (a) Wake-up-word demo “off”. (b) Wake-up-word demo “on”	493
Fig. 24.41	Example of usage of WUW technology in elevator simulator. (a) Initial state of the elevator simulator. (b) Elevator simulator acquiring voice command. (c) Elevator simulator arriving at the destination floor	495
Fig. 24.42	Executing SASE_LAB from MATLAB	496
Fig. 25.1	Depiction of the typical process involving a patient, a machine, a physician	500
Fig. 25.2	Sample recording and power spectrum of EEG signal	505
Fig. 26.1	Signal produced by earthquake	512
Fig. 27.1	Basic components of a radar system { https://www.tutorialspoint.com/radar_systems/radar_systems_overview.htm }	518
Fig. 27.2	Monostatic radar and bistatic radar http://www.telecomabc.com/b/bistatic-radar.html	519
Fig. 27.3	Basic pulse Doppler radar overview { http://doi.org/Overviewofradars.S.Cruz-PoLINEL6069 }	520
Fig. 27.4	Moving target detection http://www.jpier.org/PIERM/pier.php?paper=16122501	522
Fig. 27.5	Adaptive spectral M-subband Kalman filter in passive radar signal analysis	524
Fig. 27.6	S-band signal in time domain	525
Fig. 27.7	(a) Direct and (b) Target signal in time and frequency domain ...	525
Fig. 27.8	Spectral analysis of S-band signal for passive radar	526
Fig. 27.9	Cross-correlation of direct and target signal	527
Fig. 27.10	Ambiguity function at zero delay	528
Fig. 27.11	Adaptive spectral radar signal analysis	528
Fig. 28.1	Image 28.1.: Covid-19 cases over time in Canada (https://covid-19-canada.uwo)	537
Fig. 28.2	Image 28.2.: Cumulative confirmed cases (https://covid-19-canada.uwo)	537
Fig. 28.3	Image 28.3.: Visualization of Covid-19 cases using exploratory data analysis (https://covid-19-canada.uwo)	538
Fig. 29.1	Human image perception	542
Fig. 29.2	Digital image representation	543
Fig. 29.3	Visible wavelengths	543
Fig. 29.4	Gray Scale representation (MATLAB)	544
Fig. 29.5	RGB color space	546
Fig. 29.6	RGB Color representation of a computer image	547
Fig. 29.7	Individual RGB colors of an image	548
Fig. 29.8	Individually depicted RGB colors of an image	548

Fig. 29.9	Gray scale image representation	551
Fig. 29.10	Gaussian Smoothing Kernel	554
Fig. 29.11	Discretization of an Image and Filter Kernel	554
Fig. 29.12	Filtering operation	555
Fig. 29.13	Noisy and its decomposed (RGB) Image	557
Fig. 29.14	Median filtered image	558
Fig. 29.15	Original and histogram Equalized Image	558
Fig. 29.16	Original Image and its Decomposed versions of Red, Green and Blue (RGB) channel	559
Fig. 29.17	Original Image and Decomposed Images versions of Red, Green and Blue channel presented in Gray scale	560
Fig. 29.18	Original and “Red” Channel Equalized Image	560
Fig. 29.19	Original and “Green” Channel Equalized Image	561
Fig. 29.20	Original and “Blue” Channel Equalized Image	561
Fig. 29.21	Image of “lena” and its transpose	562
Fig. 29.22	Image of “lena” and its flipped version	562
Fig. 29.23	Copped image from the original	563
Fig. 29.24	Example of mage “translation”	564
Fig. 29.25	Example of the resized image	565
Fig. 29.26	Rotation of a pixel around an arbitrary origin by an angle θ	566
Fig. 29.27	Discrete nature of a digital picture can create problems with some transformations if not handled properly	567
Fig. 29.28	Histogram of a image of ‘lena’	568
Fig. 29.29	Abstraction levels of an Image	570
Fig. 29.30	Descriptive elements of an Image	570
Fig. 29.31	Sketch image of a house	571
Fig. 30.1	Risk visualization example in emergency situation (Eide 2012)	576
Fig. 31.1	Intelligent interactive machine for multimodal human-machine communication	584
Fig. 31.2	Multimodal autonomous interactive interface	585
Fig. 31.3	Interdisciplinary intelligent interactive multimodal communication	586
Fig. 31.4	SSDS components	587
Fig. 31.5	Chatbot for text based interaction using RASA https://rasa.com	587
Fig. 31.6	Gesture recognition for communication	588
Fig. 31.7	Object detection and recognition (Jain and Kasturi 1995)	589
Fig. 31.8	Object detection and recognition https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/	589
Fig. 31.9	Visualization	590
Fig. 31.10	Visualization and prediction using visual story telling	591
Fig. 31.11	Abstract view of a VE system (Kim 2005)	592
Fig. 31.12	Application Scenario of intelligent interactive machine	594

List of Tables

Table 1.1	Example of 4-bit number representations	26
Table 1.2	Range of values represented by a 16 bit and 32 bit DSP's	30
Table 1.3	Example of 4 bit signed magnitude number operations with 4 bit resulting number	31
Table 1.4	Example of 4 bit signed magnitude number operations with 8 bit resulting number	32
Table 1.5	Example of 4 bit signed magnitude number operations with 4 bit resulting number	32
Table 1.6	Example of 4 bit signed magnitude number operations with 8 bit resulting number	32
Table 1.7	Dynamic range and precision of 16-bit signed and unsigned integer and fractional representations	35
Table 1.8	Maximal, minimal, and precision values for integer and fractional 16-bit fixed-point representations	37
Table 1.9	Maximal, minimal, and precision values for integer and fractional 16-bit signed fixed-point representations	37
Table 16.1	A-weights in dB in frequency range from 63 to 2000 Hz	313
Table 18.1	The list of American English Phonemes	354
Table 23.1	Recognition result using 100 training samples and 25 test samples	446

Part I

Realms of Signal Processing

The very first question to address in a book on signal processing is: “What is a signal?” There are different interpretations of this term. The definitions are mainly motivated by the area in which one is interested. Examples are Digital signals (Boolean values, Integers), or Real Numbers.

These signals occur in two ways: with probability distributions that can be learned from finite observations or with probability distributions that are given. Signal processes contain information, however, typically this information is not directly measurable.

The properties of signals lead to abstractions. On a more abstract level, one can define properties of interest to applications. In evolution, living organisms have developed such levels and abstractions over time. Of course, one cannot repeat the evolution. However, it can still be studied because many of the primitive organisms are still alive today. Here we do not consider evolution but we can still make use of observations and results.

The book has three parts, going from foundations to advanced applications.

Generalities About Part I

The first part consists of 7 chapters and provides a survey of mathematical and engineering foundations assumed to be known by the reader. While the part can be described as a concise compendium, with pointers to relevant textbooks, it nevertheless presents examples and questions that can accompany a prerequisites review section of a class, or that can tune the reading engineer to the expected level of knowledge needed for the later incursion into applications.

The reader should look into these chapters to verify having the prior knowledge, or to review the concepts involved in the subsequent discussions. At a first one can skip reading this part in detail as it is sufficient to check the titles of the sections while later when the concepts are used the reader can return to

the corresponding section for refreshing one's comprehensive understanding and corresponding notations in the book.

Overview of Part I

The part starts with a review of the representations for numbers, as encountered in the representation of signals or during processing, and including the history clarifying what made representations suitable for specific applications.

Further the representation of the signals is introduced both in terms of abstract notations used for describing processes in the book, and in terms of storage in computers and other electronic devices. This introduction is accompanied by reviewing basic concepts like signal power, as well as basic processing such as convolution and correlation.

After a short review of the main transform function and of some of their properties, the concepts of linear and adaptive filters are surveyed together with major notation conventions and stability conditions.

A somewhat more in-depth review of Maximum A Posterior and Maximum Likelihood estimation of signal model parameters is subsequently presented together with motivations, examples, and related concepts.

Major adaptive filters are then surveyed, such as the Kalman and Particle Filters together with a list of their most important equations.

Part I concludes with a review of filters based on spectral analysis and involved techniques such as independent component analysis and principal component analysis.

Main Topics in Part I

A list of the main topics in Part I contains:

- Digital signal representation based on numbers,
- Quantization of Signals
- Common Signal Measurements and Features
- Central Limit Theorem
- Laplace, Fourier, Z, Cosine and Wavelet Transforms for continuous and discrete signals.
- Linear Filters and Windowing
- Transformation, processing, digital and adaptive filtering,
- Wiener, Kalman, Particle Filters and Sampling
- Estimation, detection and spectral analysis.

Chapter 1

Digital Signal Representation



To bridge the gap from theory to practice one has to master the conventions used to represent the data.

Overview

The details of digital representations of discrete-time signals are presented in this chapter bridging the gap from the abstract discrete-time signal notation presented in the book, $x[n]$, and its representation in a digital processor, and more specifically a digital signal processor.

1.1 Introduction

Continuous¹ signals are necessarily sampled at discrete time intervals as well as approximated by a finite number of discrete magnitude values to be represented digitally. Because digital processing devices process data at discrete time steps, continuous signals must be sampled at discrete time intervals. It turns out that it is possible to sample many continuous signals at discrete time intervals, producing discrete-time signals, without any loss or degradation as compared to the original analog signal. Converted continuous signals to their discrete-time representation are identical to it if certain conditions are met. Those conditions are described in the Sampling Theorem.

An additional limitation of digital processing devices, the degree of which is dictated by their architecture, is the restriction that the data must be represented by a finite number of digits, or more specifically by a finite number of bits. Typically, digital processors are designed to store and process data that have fixed specific minimal and maximal number of bits allocated for each representation. These restrictions impose representations to have a finite-precision. The process

¹ Continuous signals are referred in literature also as Analog signals. Both terms are used here interchangeably unless stated otherwise.

of representing a continuous actual value by its discrete representation is known as Quantization. When finite-precision is used to represent actual values the following steps control the quantization effects on the output.

1. Quantize in time and magnitude continuous input value $x(t)$ of the signal to obtain discrete-time sequence $x[n]$,
2. Quantize actual values of the coefficients from

$$\{A_k, k = 0, \dots, N\}$$

to a finite-precision representation:

$$\{a_k, k = 0, \dots, N\},$$

and

3. Consider the effects of arithmetic operations using finite-precision representations on the output and modify implementation as necessary to obtain an optimal result.

The effects of quantization on the continuous signal and finite-precision operations are well studied and understood. Consequently, it is possible to convert continuous signals to digital, process them, and reconstruct it back to continuous representation with desired quality. Reconstructed signals typically have characteristics that fulfill certain quality criteria that are preferred to analog counterparts. In the proceeding sections, all three numbered enumerated issues regarding the representation of data with finite-precision are discussed. However, it is also important to understand the development of the abstract concept of numbers and the historical roots of such representation. Discussion of numbers and number systems is introduced from a historical perspective that, it is believed, will shed light into on fundamental concepts of numbers and number systems that shaped current understanding of the numbers and how they are represented.

1.2 Numbers

The development of human civilization is closely followed by the development of representations of numbers. In English numbers are represented by numerals. In the past, there were several kinds of numeral notations, and symbols. In the early days, one pile of items was considered equivalent to another pile of a different number of items of a different kind. This value system was used for the trading of goods. Further development was achieved with the standardization of “value”; a fixed number of items of one kind (e.g., 5) placed in a special corresponding place, and it was considered equivalent to one item of a special kind placed in another place. This correspondence also led to earlier ways of representing numbers in written form. Since the early days, the way we do arithmetic is intimately related to the way we represent numbers.

1.2.1 *Numbers and Numerals*

As stated earlier, the development of human civilization is closely followed by the development of representations of numbers. Numbers are represented by numerals.²

1.2.1.1 Number Systems

Earlier Number Systems named after the cultures/civilizations that used it are listed below:

- Babylonian
- Egyptian
- Maya
- Greek
- Roman
- Hindu-Arabic

1.2.1.2 The Babylonian System

The earliest recorded numerals are on Sumerian clay tablets dating from the first half of the third millennium B.C. The Sumerian system was later taken over by the Babylonians. The everyday system for relatively small numbers was based on a grouping by tens, hundreds, etc. inherited from Mesopotamian civilizations. Large numbers were seldom used. More difficult mathematical problems were considered by using sexagesimal (radix 60) positional notation. Sexagesimal notation was highly developed as early as 1750 B.C. This notation was unique in that it was actually a floating-point form of representation with exponentials omitted. Proper scale factors or power of sixty was to be supplied by the context. The Babylonian cuneiform³ script was formed by impressing wedge-shaped marks in clay tablets. It is because the ancients made astronomical calculations in base 60 that we still use this system for measuring time. One hour is comprised of 60 min, 1 min of 60 s.

The circle comprises 360° degrees ($^\circ$) because the earth circles the sun in roughly 360 days. Due to Babylonians, each degree is divided into 60 min ('') and each minute into 60 s (''), and each second into 60 thirds (''''). Babylonian notation was positional (e.g., place value notation). The same symbol may mean 1, 60, 60^2 , according to its position. Since they had no concept of zero this notation could be confusing because of ambiguity.

² Webster Dictionary defines numeral as:

Function: noun

1: a conventional symbol that represents a number.

³ From Latin *cuneus*—wedge.

1.2.1.3 The Egyptian System

The Egyptian system used “|” for 1, 11“||||” for 5, \cap for 10 and $\cap\cap\cap\cap\cap$ for 50, etc. Because they used a different symbol for ones, tens, hundreds, thousands, etc. the range of numbers that could be represented was limited. Note that later Romans adopted this system to represent their numbers.

1.2.1.4 Maya Indians

From ancient civilizations, only Maya Indians have used the concept of “zero” as a quantity sometime around 200 A.D. They have also introduced fixed-point notation as early as first century A.D. Their number system was a radix-20 system.

1.2.1.5 The Greek System: Abacus

Greek numerals from about the fifth century B.C. used alphabetic characters (24 characters) to represent numbers. Since 27 symbols were needed three letters of Semitic origin were adopted. Greek Abacus originates at about the + century B.C. Row and Columns of pebbles are organized in a matrix that corresponds to our decimal system. The written form did not follow the positional notation of the decimal system. On the other hand, Greek astronomers make use of a sexadecimal positional notation for fractions, adapted from Babylonians.

1.2.1.6 Roman System

Because Roman numerals were in use in Europe for over a thousand years, we are still familiar with them and use them in certain instances (clock faces, enumerated lists in written documents, monuments, etc.). The Roman number system was based on Etruscan letter notations *I*, *V*, *X*, *L*, *C*, *D*, and *M* for 1, 5, 10, 50, 100, 500, 1000. The subtractive principle, whereby 9 and 40 are written as *IX* and *XL*, became popular during medieval times since it was hardly used by the Romans. It is interesting to note that the original symbol for *M* (1000) was ∞ . The symbol ∞ is a corruption of symbol (approximately) ω . In 1655 John Wallis proposed that this symbol, ∞ , be used for “infinity”.

1.2.1.7 Hindu-Arabic Numerals

The numeration we use now: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, is often referred to as Arabic notation, but it is of Hindu origin. It is transmitted to Europe by Arab scholars. The value of a digit depends on its position in the system (its place in the number determines its value). Consequently, zero is needed to be able to represent

numbers unambiguously. For example, 704 compared to 74. It was this way that the concept of zero was forced itself onto Indian mathematicians. In theory, zero is also needed occasionally in the Babylonian system, but as the base is much larger, the context would usually supply the missing information. Consequently, Babylonians struggled on without zero for over a thousand years.

Such earlier notations were inconvenient for performing arithmetic operations except for the simplest cases. Analysis of those earlier number systems also reveals two distinct approaches: **sign-value** notation (e.g., Roman Numeral System) and **positional** notation or **place-value** notation that is commonly used today. Furthermore, the abstract concept of a number and the objects being counted were not separable for a long time as exemplified by many languages. In those languages there are many names for a number of particular objects but not for the idea of numbers. For example, Fiji Islanders use “**bolo**” for ten boats, but “**koro**” for ten coconuts. In the English language, couple refers to two people, a century to 100 years, etc.

1.2.2 Types of Numbers

To understand how the numbers, e.g. numerals, are represented in modern digital computing systems it is important to know what kind of possible numbers are in use.

1.2.2.1 Whole Numbers

The whole numbers are $1, 2, 3, 4, \dots$, defining the set \mathbb{N} , also called the counting or natural numbers. The number 0 is sometimes included in the list of “whole” numbers, but there seems to be no general agreement. Some authors also interpret “the whole number” to mean “a number having fractional part of zero,” making the whole numbers equivalent to the integers.

1.2.2.2 Integer Numbers

Advancement of mathematics brought by the discipline of algebra forced the recognition of negative numbers (e.g., to obtain the solution of the following equation $2x + 9 = 3$ requires the introduction of negative numbers). The set of whole numbers when extended with zero and negative whole numbers defines the set \mathbb{Z} of integers: $\dots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \dots$

1.2.2.3 Fractions or Rational Numbers

A fraction of a rational number is defined as the ratio of two whole numbers p , q : $\frac{p}{q}$. The set of all rational numbers is denoted with \mathbb{Q} derived from the German word Quotient which can be translated as a ratio. Most of the early systems used and named only a few obvious common fractions. In the famous Rhind papyrus,⁴ a famous document from the Egyptian Middle Kingdom that dates to 1650 BC, only simple names for the unit fractions were used: $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, $\frac{1}{5}$, ..., and for $\frac{2}{3}$. Other fractions when required were obtained by adding these simple fractions. For example: $\frac{5}{7} = \frac{1}{2} + \frac{1}{7} + \frac{1}{14}$.

1.2.2.4 Irrational Numbers

The Discovery of irrational numbers is attributed to Pythagoras, who found that the diagonal of a square is not a rational multiple of its side of the square. With sides equal to 1—e.g., $diagonal = \sqrt{2}$. In other words, the ratio of diagonal to the side cannot be expressed by whole numbers. Irrational numbers have decimal expansions that neither terminate nor become periodic. Examples of irrational numbers are $\sqrt{2}$, $\sqrt{3}$, π , e .

1.2.2.5 Real and Complex Numbers

The collection of rational and irrational numbers defines the set \mathbb{R} of real numbers. Real numbers can be extended to complex numbers with the addition of imaginary numbers $i = \sqrt{-1}$. A complex number z is expressed as:

$$z = x + iy$$

1.2.3 Positional Number Systems

In the positional notation, the value of a number depends on the numeral as well as its position within the number. Typically, the value of the position is the power of ten. For example, the number represented by the numeral 1957 is equal to 7 ones, 5 ten's, 9 hundred's, and 1 thousand's. This concept leads to the generalization of the value represented by a numeral as follows:

⁴ It was found in the memorial temple (or mortuary temple) of Pharaoh Ramesses II.

$$x = \pm(d_{n-1}B^{n-1} + \cdots + d_1B^1 + d_0B^0 \cdot d_{-1}B^{-1} + \cdots + d_{-m}B^{-m})$$

where: \pm is the sign of the number, $d_i \in \{0, 1, 2, \dots, B-1\}$ is the set of numerals, “ \cdot ” is decimal, or in general radix point, and B is the base of the number system. Note that the number to the left of the radix point, called the integral part, denotes an integer part of the number represented by n numerals. The number to the right of the radix point called a fractional part represents a fractional number less than 1 represented by m numerals. With this notation, the set of real numbers \mathbb{R} can be represented.

Computers can only use a finite subset of the numbers due to finite resources available to represent a number. Consequently, only a finite and limited set of numbers can be represented. This set is defined by the total number of elements that it can represent as well as the range of values that it covers. The most common native representation of a numeral in a computer is in the Binary system or base $B = 2$. The numerical value in our accustomed, reference base 10, number system of a base 2 (or binary) number is given by the following expression:

$$x = \pm(d_{n-1}2^{n-1} + \cdots + d_12^1 + d_02^0 \cdot d_{-1}2^{-1} + \cdots + d_{-m}2^{-m})$$

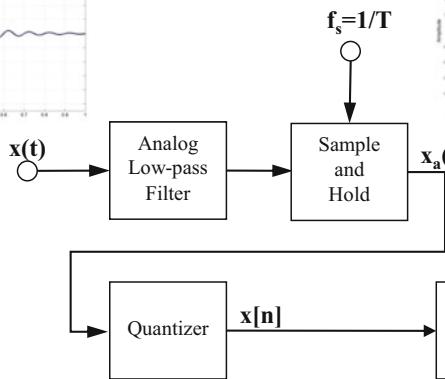
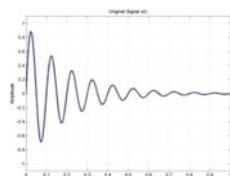
where: \pm is the sign of the number, $d_i \in \{0, 1\}$ takes values from the set of binary numerals, and “ \cdot ” is the binary point. The range of values and their precision is defined by n , the number of bits used to represent the integer portion of a number, and m , the number of bits to represent the fractional part of the number.

1.3 Sampling and Reconstruction of Signals

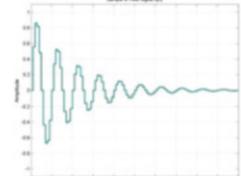
Typical system interfaces with the continuous world via Analog-to-Digital (ADC) and Digital-to-Analog (DAC) Converters as depicted in Fig. 1.1.

A signal is identically reconstructed from discrete samples only when conditions specified by the so called Sampling Theorem are satisfied. Otherwise, the existence of a difference between the reconstructed signal and the original is called aliasing. To satisfy Sampling Theorem requirements, the continuous input signal must be ensured to be band-limited. Thus, the ADC is preceded by a low-pass filter. This pre-filtering is a critical step in any digital processing system. It ensures that the effects of aliasing are minimized to levels that are not perceptible by the intended audience. The filter is implemented as an analog low-pass filter. The band-limited signal is then sampled by a fixed sample rate or sampling frequency, f_s . The sampling is performed by a sample-and-hold device. This signal is then quantized and represented in a digital form as a sequence of binary digits/bits that have values of 1's and 0's. Quantized representation of the data is then converted to a desired digital representation of a DSP to facilitate further processing. The conversion process is depicted in Fig. 1.1.

a) Continuous Signal



b) Amplitude Quantized Signal



c) Amplitude & Time Quantized – Digital Signal

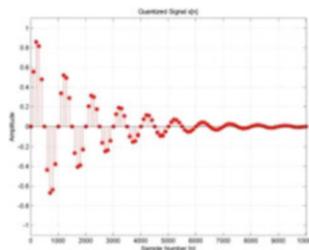


Fig. 1.1 The interfacing with a continuous signal that is optionally conditioned by the sensor conditioner. Analog-to-Digital Conversion: (a) Continuous signal $x(t)$, (b) Sampled signal $x_a(t)$ with sampling period T satisfying Nyquist rate as specified by Sampling Theorem, (c) Digital sequence $x[n]$ obtained after sampling and quantization

Example 1.1 Assume that the input continuous-time signal is a pure periodic signal represented by the following expression:

$$x(t) = A \sin(\omega_0 t + \phi) = A \sin(2\pi f_0 t + \phi)$$

where A is amplitude of the signal, ω_0 is frequency in radians per second (rad/sec), ϕ is phase in radians, and f_0 is frequency in cycles per second measured in Hertz (Hz). Assuming that the continuous-time signal $x(t)$ is sampled every T seconds or alternatively with the sampling rate of $f_s = 1/T$, the discrete-time signal $x[n]$ representation obtained by $t = nT$ will be:

$$x[n] = A \sin(\omega_0 nT + \phi) = A \sin(2\pi f_0 nT + \phi)$$

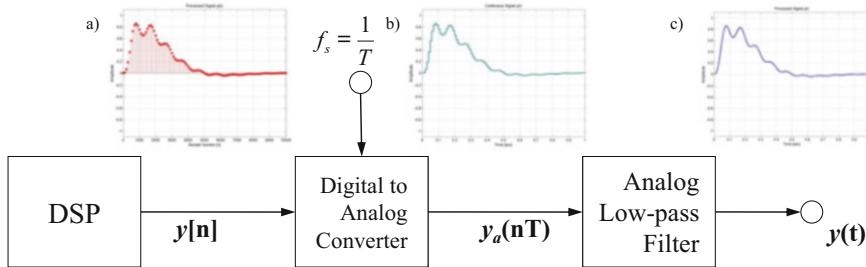


Fig. 1.2 Digital-to-analog conversion. (a) Processed digital signal $y[n]$. (b) Continuous signal representation $y_a(nT)$. (c) Low-pass filtered continuous signal $y(t)$

Alternative representation of $x[n]$:

$$x[n] = A \sin\left(2\pi \frac{f_0}{f_s} n + \phi\right) = A \sin\left(2\pi F_0 n + \phi\right) = A \sin(\Omega_0 n + \phi)$$

reveals additional properties of the discrete-time signal. The $F_0 = f_0/f_s$ defines **normalized frequency**, and Ω_0 **digital frequency** where:

$$\Omega_0 = 2\pi F_0, \quad 0 \leq \Omega_0 \leq 2\pi$$

A **DSP** processor performs a programmed operation, typically a complex algorithm, on the suitably represented input signal. The result is obtained as a sequence of digital values. Those values after being converted into an appropriate data representation (e.g., 24 bit signed integers) are converted back into continuous domain via **digital-to-analog** converter: **DAC**. The procedure is depicted in Fig. 1.2.

Quantization in time, via sampling, as well as in amplitude of continuous input signals, $x(t)$, to discrete-time signal $x[n]$, as well as coefficients of digital signal processing structures requires also resolving how the numbers are represented by a digital signal processor.

The next section will discuss issues of quantization, numbers, and their representations.

1.3.1 Scalar Quantization

The component of the system that transforms an input value $x[n]$ into one of a finite set of prescribed values $\hat{x}[n]$ is called scalar quantization. As depicted in Fig. 1.3, this function is depicted with ideal sample-and-hold followed by Analog to Digital Converter. This function can be further refined by the representation depicted in Fig. 1.3. The ideal **C/D** converter represents the sampling performed by the sample-and-hold, and quantizer and coder combined represent **ADC**.

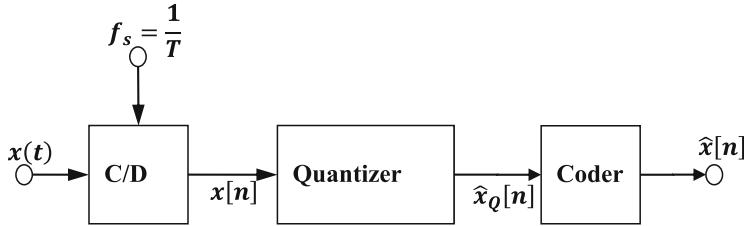


Fig. 1.3 Conceptual representation of ADC

This conceptual abstraction allows us to assume that the sequence $x[n]$ is obtained with infinite precision. Those values $x[n]$ are scalar quantized to a set of finite precision amplitudes denoted hereby $\hat{x}_Q[n]$. Furthermore, quantization allows that this finite-precision set of amplitudes to be represented by a corresponding set of (bit) patterns or symbols, $\hat{x}[n]$. Without loss of generality, it can be assumed that input signals cover a finite range of values defined by minimal, x_{min} , and maximal values, x_{max} , respectively. This assumption in turn implies that the set of symbols representing $\hat{x}[n]$ is finite. The process of representing a finite set of values to a finite set of symbols is known as *encoding* performed by the coder, as in Fig. 1.3. Thus one can view quantization and coding as a mapping of the infinite precision value of $x[n]$ to a finite precision representation $\hat{x}[n]$ picked from a finite set of symbols.

Quantization, therefore, is a mapping of a value $x[n]$, $x_{min} \leq x \leq x_{max}$, to $\hat{x}[n]$. The quantizer operator, denoted by $Q(x)$, is defined by:

$$\hat{x}[n] = \hat{x}_i = Q[x[n]], \quad x_{i-1} < x[n] \leq x_i$$

where $\hat{x}[n]$ denotes one of L possible quantization levels where $1 \leq i \leq L$ and x_i represent one of $L + 1$ decision levels. The above expression is interpreted as follows; If $x_i < x[n] \leq x_i$, then $x[n]$ is quantized to the quantization level \hat{x}_i and $x[i]$ is considered a quantized sample of $x[n]$.

Clearly from the limited range of input values and a finite number of symbols it follows that quantization is characterized by its quantization step size Δ_i defined by the difference of two consecutive decision levels:...

$$\Delta_i = x_i - x_{i-1}$$

Example 1.2 Uniform quantization of L -levels covering the range from x_{min} to x_{max} ,

$$\Delta = \Delta_i = \frac{(x_{max} - x_{min})}{L} \quad i = 0, \dots, L - 1$$

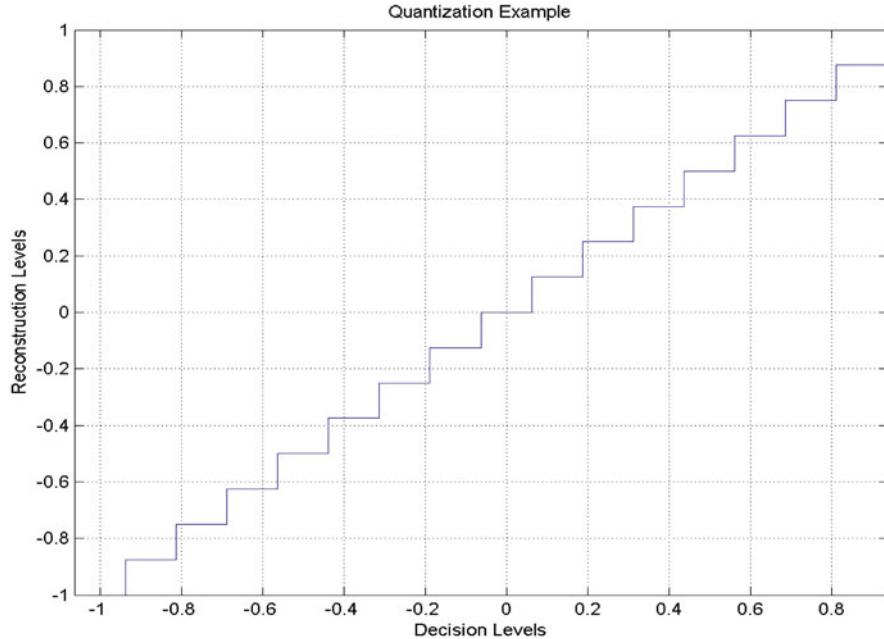


Fig. 1.4 Example of a uniform quantization for $L = 16$ levels. As it is discussed in the following sections, $L = 16$ levels, require 4 bits of the codeword to represent each level. Because the distribution of the input values is uniform the decision and reconstruction levels are uniformly spaced: $\hat{x} = Q(x)$

- Decision Levels:

$$\left[-\frac{15\Delta}{2}, -\frac{13\Delta}{2}, -\dots, -\frac{3\Delta}{2}, -\frac{1\Delta}{2}, +\frac{1\Delta}{2}, +\frac{3\Delta}{2}, -\dots, +\frac{13\Delta}{2}, +\frac{15\Delta}{2} \right]$$

- Reconstruction Levels:

$$[-8\Delta, -7\Delta, -\dots, -3\Delta, -2\Delta, -1\Delta, +1\Delta, +2\Delta, +3\Delta, -\dots, +7\Delta, +8\Delta]$$

In the previous example, namely Fig. 1.4, a uniform quantizer was described. Here a uniform quantizer is formally defined as one whose decision and reconstruction levels are uniformly spaced. Specifically:

$$\Delta = \Delta_i = x_i - x_{i-1}, \quad \& i \leq i \leq L$$

$$\hat{x} = \frac{x_i + x_{i-1}}{2}, \quad \& i \leq i \leq L$$

Thus, Δ , the step size equal to the spacing between any two consecutive decision levels, is constant for any two consecutive reconstruction levels in a uniform quantizer. Each reconstruction level has attached a *symbol*—or the *codeword*. Binary numbers are typically used to represent the quantized samples. The term *Codebook* refers to a collection of all codewords or symbols. In general, with B -bit binary codebook there are 2^B different quantization (or reconstruction) levels. This representational issue is detailed in the following sections.

When designing or applying a uniform scalar quantizer, the knowledge of the maximum value of the sequence is required. Typically the range of the input signal (e.g., speech, audio, video), is expressed in terms of the standard deviation, σ_x of the probability density function (*pdf*) of the signals' amplitudes. Specifically, it is often assumed that the range of input values is equal to $-4\sigma_x \leq x[n] \leq 4\sigma_x$ where σ_x is signal's standard deviation.

In addition to quantization, many algorithms depend on accurate yet simple mathematical models describing statistics of signals. Several studies have been conducted on speech signals assuming that speech signal amplitudes are realizations of a random process. The accuracy of several models was evaluated as function duration of speech segment used for capturing speech statistics (Jensen 2005).

The following functions, also depicted in Fig. 1.5, are evaluated as models of speech signal *pdf*:

- *Gamma Distribution*

$$f(x) = \left(\frac{\sqrt{3}}{8\pi\sigma_x|x|} \right)^{\frac{1}{2}} e^{-\left(\frac{\sqrt{3}|x|}{2\sigma_x} \right)}, \quad -\infty < x < +\infty$$

- *Laplacian Distribution*

$$f(x) = \left(\frac{1}{\sqrt{2}\sigma_x} \right) e^{-\left(\frac{2|x|}{\sigma_x} \right)}, \quad -\infty < x < +\infty$$

- *Gaussian Distribution*

$$f(x) = \left(\frac{1}{\sqrt{2\pi}\sigma_x} \right) e^{-\left(\frac{x^2}{2\sigma_x^2} \right)}, \quad -\infty < x < +\infty$$

where in all three equations σ_x —is Standard Deviation.

Example 1.3 Assume a $B - bit$ binary codebook having thus $2B$ codewords or symbols. Maximum signal value is set to $x_{max} = 4\sigma_x$. What is the quantization step size of a uniform quantizer?

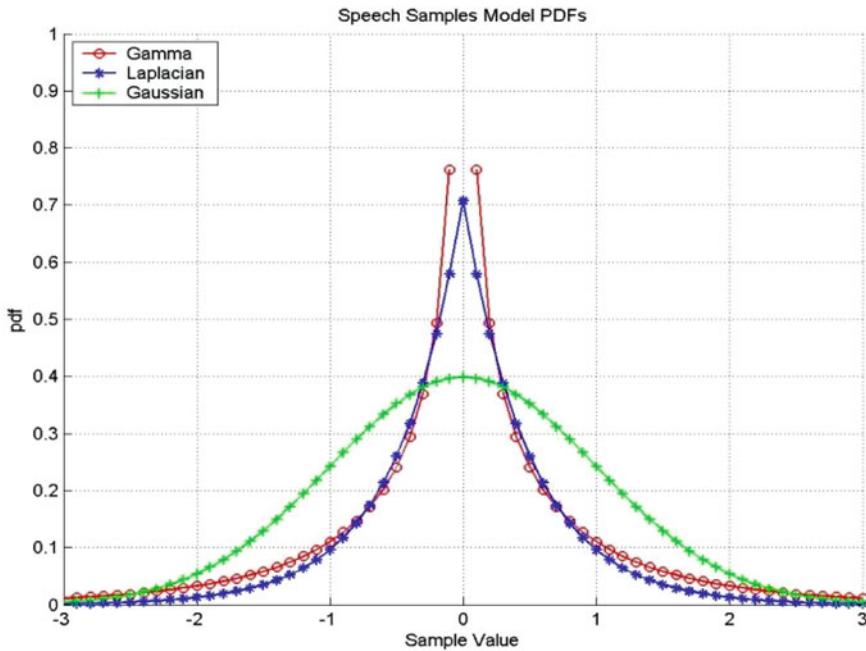


Fig. 1.5 Example of pdf's (probability distribution function)

$$\frac{2x_{max}}{\Delta} = 2^B \Rightarrow 2x_{max} = \Delta 2^B \Rightarrow \Delta = \frac{2x_{max}}{2^B}$$

From the discussion presented thus far, it is clear that the quality of representation is related to the step size of the quantizer, Δ , which in turn depends on the number of bits B used to represent a signal value. The quality of quantization typically is expressed as a function of the step size Δ and relates directly to the notion of *quantization noise*.

1.3.2 Quantization Noise

There are two classes of quantization noise:

- *Granular Distortion*
- *Overload Distortion*

1.3.2.1 Granular Distortion

Granular distortion occurs for the values of $x[n]$, unquantized signal, which falls within the range of the quantizer $[x_{min}, x_{max}]$. The quantization noise, $e[n]$, is the error that occurs because infinite precision value $x[n]$ is approximated with a finite-precision value of quantized representation \hat{x} . Specifically, quantization error $e[n]$ is defined as a difference of quantized value \hat{x} from true value $x[n]$:

$$e[n] = \hat{x}[n] - x[n]$$

For a given step size Δ the magnitude of the quantization noise $e[n]$, can be no greater than $\Delta/2$, that is:

$$-\frac{\Delta}{2} \leq e[n] \leq +\frac{\Delta}{2}$$

Example 1.4 For the periodic sine-wave signal use 3-bit and 8-bit quantizer values. The input periodic signal is given with the following expression:

$$x[n] = \cos(\omega_0 t), \quad \omega_0 = 2\pi F_0 = (2\pi)x(0.76)$$

MATLAB *fix* function is used to simulate quantization. The following figure, Fig. 1.6, depicts the results of the analysis.

Overload Distortion

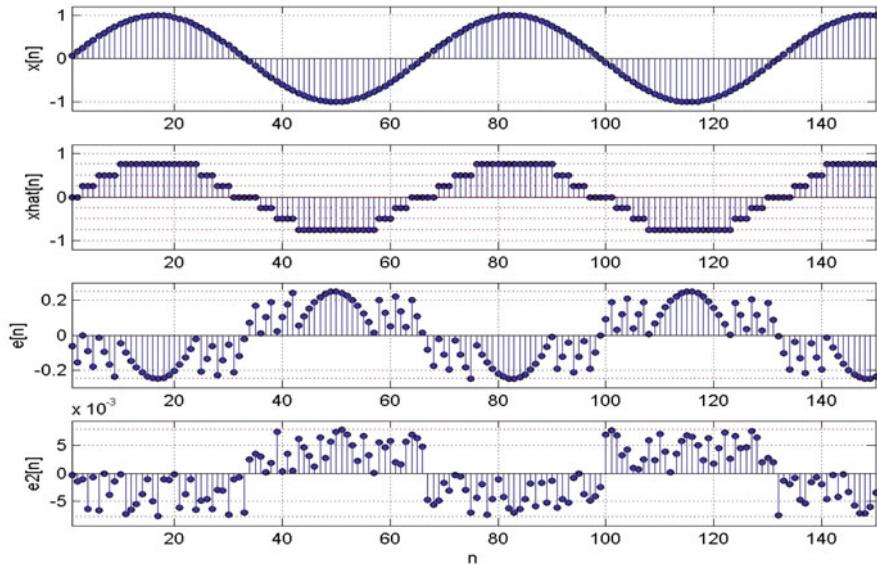
Overload distortion occurs when the samples fall outside the range covered by the quantizer. Those samples are typically *Gclipped* and they incur a quantization error in excess of $\dots \Delta/2$. Due to the small number of clipped samples, it is common to neglect the infrequent large errors in theoretical calculations.

Often the goal of signal processing in general and specifically audio or image processing is to maintain the bit rate as low as possible while maintaining a required quality level. Meeting those two criteria requires fulfilling competing requirements.

Analysis of Quantization Noise

The desired approach in analyzing the quantization error in numerous applications is to assume quantization error is an ergodic white-noise random process. This implies that the process, that is quantization error $e[n]$, is uncorrelated. In addition, it is also assumed that the quantization noise and the input signal are uncorrelated, i.e., $E(x[n]e[n+m]) = 0, \forall m$. The final assumption is that the *pdf* of the quantization noise is uniform over the quantization interval:

$$p(e) = \begin{cases} \frac{1}{\Delta}, & -\frac{\Delta}{2} \leq e \leq +\frac{1}{\Delta} \\ 0 & otherwise \end{cases}$$



1. represents sequence $x[n]$ with infinite precision,
2. represents quantized version $\hat{x}[n]$,
3. represents quantization error $e[n]$ for $B = 3$ bits ($L = 8$ quantization levels), and
4. is quantization error for $B = 8$ bits ($L = 256$ quantization levels)

Fig. 1.6 Subplots are enumerated from top to bottom

Stated assumptions are not always valid. Consider a slowly varying input signal $x[n]$, then quantization error $e[n]$ is also changing slowly, thus being signal-dependent as depicted in Fig. 1.7. Furthermore, correlated quantization noise can be annoying (e.g., image sequences—tv, or audio).

As illustrated in Fig. 1.7, when the quantization step Δ is small then assumptions for the noise being uncorrelated with itself and the signal are roughly valid particularly when the signal fluctuates rapidly among all quantization levels. In this case, quantization error approaches a white-noise process with an impulsive autocorrelation and flat spectrum.

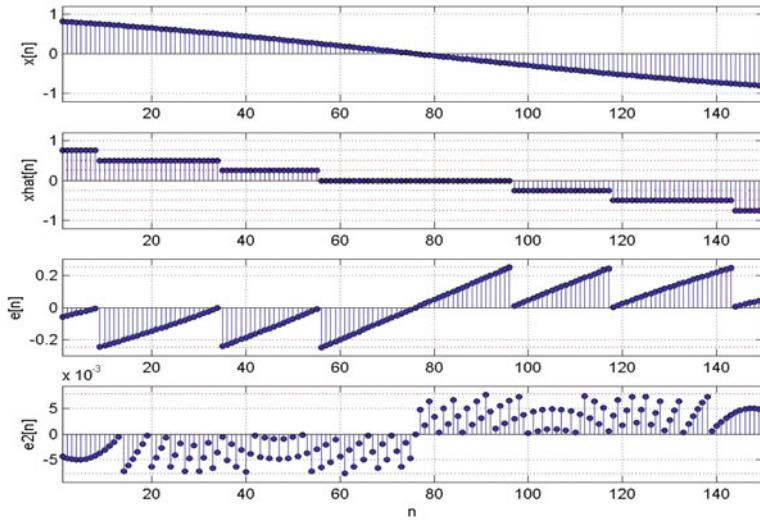
The Fig. 1.8 demonstrates quantization effects on the speech signal.⁵ As depicted in the Fig. 1.9, with increase in the number of quantization levels L , decrease of correlation can be observed as flattening of the distribution, approaching to a uniform distribution.

Depicted plots show distribution of quantization errors with:

1. $L = 2^3$,
2. $L = 2^8$ and
3. $L = 2^{16}$

quantization levels.

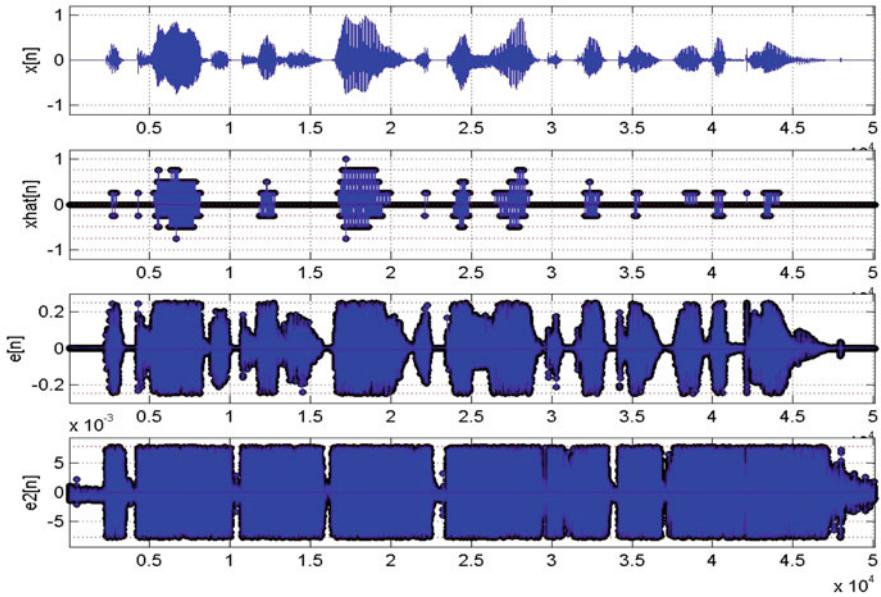
⁵ The signal was taken from the file: TEST/DR3/FPKT0/si1538.wav of the TIMIT corpus.



1. represents sequence $x[n]$ with infinite precision,
2. represents quantized version $\hat{x}[n]$,
3. represents quantization error $e[n]$ for $B = 3$ bits ($L = 9$ quantization levels), and
4. is quantization error for $B = 8$ bits ($L = 256$ quantization levels). Note the reduction in correlation level with an increase of a number of quantization levels which implies a decrease of step size δ .

Fig. 1.7 Example of slowly varying signal that causes quantization error to be correlated

As depicted in the Fig. 1.9, with increase in number quantization levels L , decrease of correlation marked as flattening of the distribution approaching to a uniform can be observed. An additional approach can be used to force $e[n]$ to be white-noise and uncorrelated with $x[n]$, namely by adding white-noise to $x[n]$ prior to quantization. The effect of this approach is demonstrated in the Fig. 1.10, obtained by adding an insignificant amount of Gaussian noise with zero mean and variance of 5 to the original signal. Dramatic improvement is clearly visible particularly for the $L = 2^{16}$ quantization level by comparing distributions with the one in Fig. 1.9. The process of adding white noise is known as *Dithering*. This de-correlation technique was shown to be useful not only in improving the perceptual quality of the quantization noise of speech signals but also with image signals.



1. represents sequence $x[n]$ with infinite precision,
2. represents quantized version $\hat{x}[n]$,
3. represents quantization error $e[n]$ for $B = 3$ bits ($L = 8$ quantization levels), and
4. is quantization error for $B = 8$ bits ($L = 256$ quantization levels). Note the reduction in correlation level with an increase of a number of quantization levels which implies a decrease of step size Δ

Fig. 1.8 Example of speech signal demonstrating the effect of step size to the degree of correlation of quantization error

1.3.3 Signal-to-Noise Ratio

A measure to quantify the severity of the quantization noise is the [Signal to Noise Ratio \(SNR\)](#). It relates the strength of the signal to the strength of the quantization noise, and it is formally defined as:

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{E[x^2[n]]}{E[e^2[n]]} \approx \frac{\frac{1}{N} \sum_{n=0}^{N-1} x^2}{\frac{1}{N} \sum_{n=0}^{N-1} e^2}$$

Given the following assumptions:

- Quantizer range $2x_{max}$, and
- Quantization interval: $\Delta = \frac{2x_{max}}{2^B}$, for a B -bit quantizer, and
- Uniform pdf of the quantization error $e[n]$,

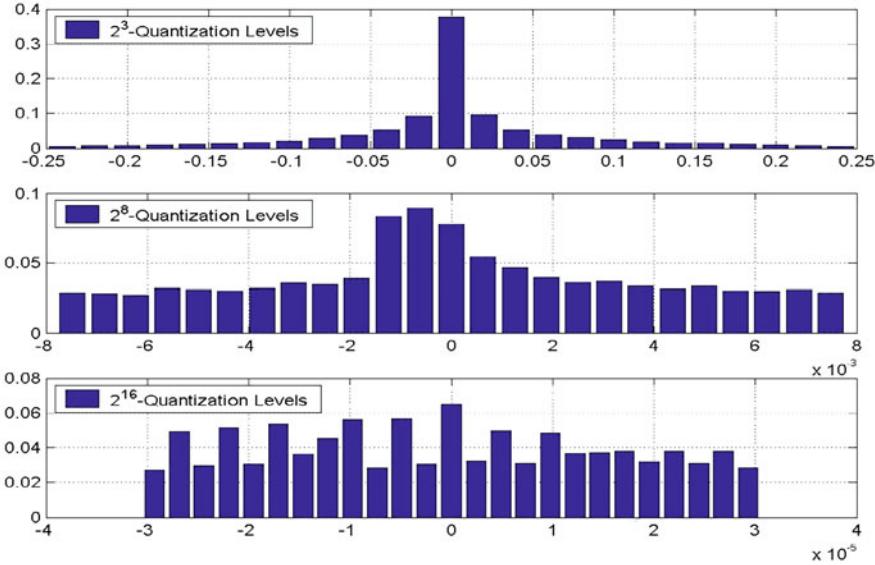


Fig. 1.9 Histogram of quantization error for speech signal. Note the reduction of error magnitude as well as increase of uniformity of the distribution with increase of number of quantization levels

it can be shown that

$$\sigma_x^2 = \frac{\Delta^2}{12} = \frac{\left(\frac{2x_{max}}{2^B}\right)^2}{12} = \frac{x_{max}^2}{(3)2^{2B}}$$

Thus SNR can be expressed as:

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \sigma_x^2 \left(\frac{(3)2^{2B}}{x_{max}^2} \right)$$

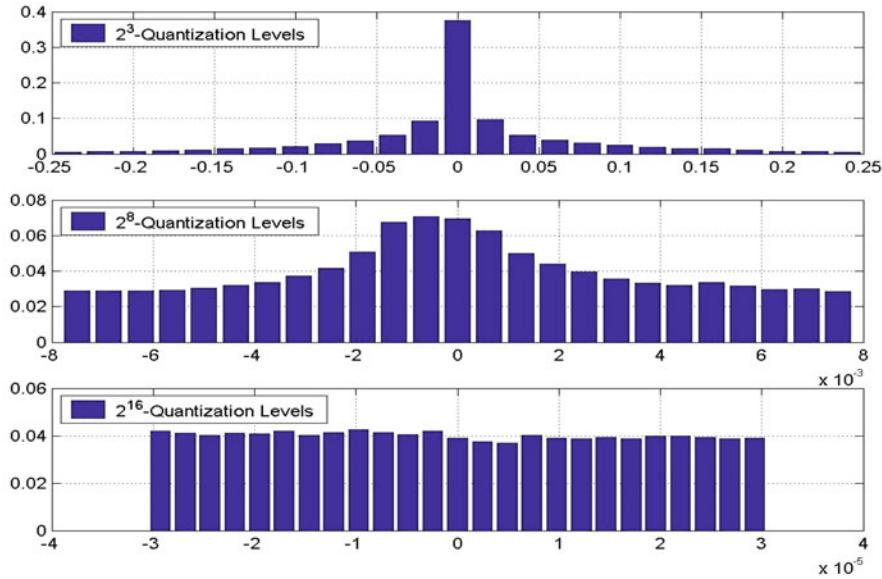
or in decibels dB can be written as:

$$SNR(dB) = 10 \left(\log_{10} 3 + 2B \log_{10} 2 \right) - 10 \log_{10} \left(\frac{x_{max}^2}{\sigma_x^2} \right) \approx 6B + 4.77 - 20 \log_{10} \frac{x_{max}}{\sigma_x}$$

Assuming that maximal value x_{max} , obtained from the pdf of the distribution of $x[n]$, is set to $x_{max} = 4\sigma_x$, then $SNR(dB)$ expression becomes:

$$SNR(dB) \approx 6B - 7.2$$

Example 1.5 For uniform quantizer with the quantization interval Δ , derive the variance of the error signal. Consider that the signal is random with uniform probability distribution within the interval defined by Δ as defined in the figure below (Fig. 1.11).



Plots depict distribution of quantization errors with:

1. $L = 2^3$,
2. $L = 2^8$ and
3. $L = 2^{16}$ quantization levels.

Note the reduction of error magnitude as well as increase of uniformity of the distribution with increase of number of quantization levels.

Fig. 1.10 Histogram of quantization error for speech signal after adding Gaussian noise with zero mean and variance of 5

The mean and variance of the $p(e)$ are the first two moments, m , of the random process defined as the expected value of random variable e :

$$E(e^m) = \int_{-\infty}^{+\infty} e^m p(e) de$$

Thus mean and variance of the $p(e)$ are:

$$m_e = \int_{-\frac{\Delta}{2}}^{+\frac{\Delta}{2}} \frac{1}{\Delta} de = 0$$

$$\sigma_e^2 = \int_{-\frac{\Delta}{2}}^{+\frac{\Delta}{2}} \frac{1}{\Delta} e^2 de = 2 \frac{1}{\Delta} \int_0^{\frac{\Delta}{2}} e^2 de = \frac{2}{3\Delta} \left(\frac{\Delta}{2}\right)^3 = \frac{\Delta^2}{12}$$

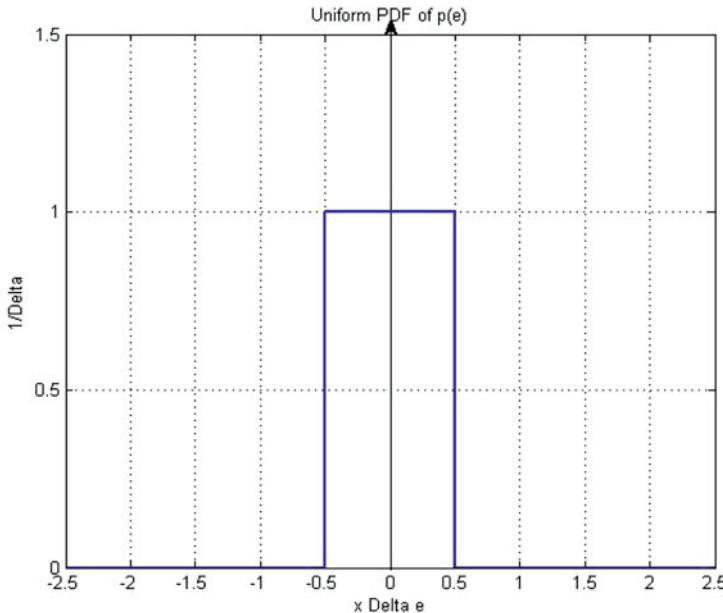


Fig. 1.11 Uniform probability distribution function $p(e)$ of error signal in the range $[-\Delta/2, \Delta/2]$

1.3.4 Transmission Rate

Another important factor in utilizing the *DSP* processors is the **Bitrate R** . Bit rate is computed with the following expression where f_s sample rate in Hz or samples per second, and B is the number of bits used to represent a sample:

$$R = Bf_s$$

The presented quantization scheme is called *pulse code modulation* (PCM), where B -bits per sample are transmitted as a codeword. Advantages of this scheme are:

- It is *instantaneous* (no coding delay)
- It is *independent* of the signal content (voice, music, etc.)

Disadvantages:

- It requires a high bit rate for good quality.

Example 1.6 For “toll quality” (equivalent to a standard telephone quality) of a signal minimum of 11 bits per sample is required. For 10,000 Hz sampling rate, the required bit rate is: $B = (11 \text{ bits/sample}) \times (10,000 \text{ samples/sec}) = 110,000 \text{ bps} = 110 \text{ kbps}$. For *CD*-quality signal with a sample rate of 20,000 Hz and 16-bits/sample, $\text{SNR} (\text{dB}) = 96 - 7.2 = 88.8 \text{ dB}$ and bit rate of 320 kbps.

Because sampling rate is fixed for most applications this goal implies that the bit rate is reduced by decreasing the number of bits per sample. This area is of significant importance for communication systems and is known as **Coding** (Jayant and Noll 1984; Conway and Guy 1999). However, this coding refers to the information encoding procedures beyond the representation of the numerical values that are being discussed here. However, as indicated earlier, uniform quantization is optimal only if the distribution of input samples $x[n]$ is uniform. Thus, uniform quantization may not be optimal in general— SNR cannot be as small as possible for a certain number of decision and reconstruction levels.

Consider, for example, a speech signal for which $x[n]$ is much more likely to be in one particular region than in another (low values occurring much more often than the high values), as exemplified by Fig. 1.5. This implies that decision and reconstruction levels are not being utilized effectively with uniform intervals over $\cdots x_{max}$. An optimal solution must account for the distribution of input samples.

1.3.5 Nonuniform Quantizer

A quantization that is optimal (in a least-squared error sense) for a particular *pdf* is referred to as the *Max Quantizer*. For a random variable x with a known *pdf*, it is required to find the set of M quantizer levels that minimizes the quantization error. Therefore, finding the decision and reconstruction levels x_i and \hat{x}_i , respectively, that minimizes the mean-squared error (*MSE*) distortion measure:

$$D = E \left[(x_i - \hat{x}_i)^2 \right]$$

E-denotes expected value and \hat{x}_i is quantized version of X_i , would give us optimal decision levels. It turns out that optimal decision levels are given by the following expression:

$$x_k = \frac{\hat{x}_{k+1} + \hat{x}_k}{2}, \quad 1 \leq k \leq L - 1$$

On the other hand, the optimal reconstruction level x_k is the centroid of $p_x(x)$ over the interval $x_{k-1} \leq x \leq x_k$ computed by the following expression:

$$\hat{x}_k = \int_{x_{k-1}}^{x_k} \left[\frac{p_x(x)}{\int_{x_{k-1}}^{x_k} p_x(x') dx'} \right] x dx = \int_{x_{k-1}}^{x_k} \tilde{p}_x(x) dx$$

The above expression is interpreted as the mean value of x over interval $x_{k-1} \leq x \leq x_k$ for the normalized *pdf* $\tilde{p}_x(x)$.

Solving the last two equations for x_k and \hat{x}_k is a nonlinear problem in these two variables. There is an iterative solution that requires obtaining *PDFs* of x ; an accurate estimate of which can be difficult.

1.3.6 Companding

The idea behind *companding* is based on the fact that a uniform quantizer is optimal for a uniform *pdf*, thus, if a nonlinearity transformation T is applied to unquantized input $x[n]$ to form a new sequence $y[n]$ whose *pdf* is uniform. A uniform quantizer can be applied to $y[n]$ to obtain \hat{y} .

A companding operation compresses the dynamic range of input samples for encoding and expands the dynamic range on decoding. Optimal application of companding procedure requires accurate estimation of *pdf* of input $x[n]$ values from which non-linear transformation T can be derived. In practice, however, such transformations are standardized under CCITT (CCITT) international standard coder at 64 kbps; specifically $A - law$ and $\mu - law$ companding. $A - law$ is used in Europe while $\mu - law$ in North America.

The $\mu - law$ transformation is given by:

$$T(x[n]) = x_{max} \frac{\log \left(1 + \mu \frac{|x[n]|}{x_{max}} \right)}{\log 1 + \mu} \text{sign}(x[n])$$

The $\mu - law$ transformation for $\mu 255$, North American *PCM* standard, is followed by 8-*bit* uniform quantization, 7-*bits* for value, and 1-*bit* for a sign, achieves “toll quality of speech” in telephone channels. Achieved toll-quality is equivalent quality to straight uniform quantization using 12 bits.

Due to standardization, in digital telephone networks and voice modems, standard *CODEC*⁶ chips are used in which audio is digitized in an 8-*bit* format (Fig. 1.12).

1.4 Data Representations

The *Digital Signal Processor (DSPs)*, similarly to general computer processors, support several data formats. The variety of data formats and computational operations determine *DSP* capabilities. The most general classification of *DSP* processors is in terms of their hardware support of data types for various operations (e.g., addition, subtraction, multiplication, and division). *DSP*’s are thus categorized

⁶ The word *CODEC* is derived from CODer-DECoder.

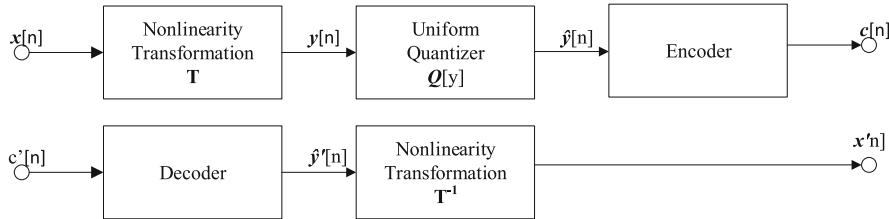


Fig. 1.12 Block diagram of companding in the transmitting and receiving *DSP* system. The $x[n]$ is an unquantized input sample with a nonuniform *pdf* of values; $y[n]$ is the value obtained after nonlinear transformation with uniform *pdf* values, and $\hat{y}[n]$ is quantized sample, $c[n]$ is encoded binary representation of this sample value. This binary encoded stream is typically transmitted to a receiving system where it is converted back to the original by decoding the input encoded samples $c'[n]$ to $\hat{y}'[n]$, and applying the inverse of the non-linear transformation T^{-1} obtaining sequence $x'[n]$. If $c'[n] = c[n]$, then $x'[n]$ differs from $x[n]$ by amount of introduced quantization noise

as *fixed-point* or *floating-point* devices. Fixed-point data are computer representations of integer numbers. Floating-point data types are computer representations of real numbers.

1.4.1 Fixed-Point Number Representations

In mathematics, the range of values that an integer number can take is unlimited. That is, an integer number can take values ranging from $-\infty$ to $+\infty$:

$$-\infty, \dots, -3, -2, -1, 0, 1, 2, 3, \dots, +\infty$$

Due to the limitations of the hardware, the integer representations in a computer are restricted to a range that is directly dependent on the number of bits of memory available, or the number of bits allocated for numbers. For example, if a processor uses 4 bits to represent a number, there are a total of $2^4 = 16$ possible distinct combinations directly supported in hardware. If one would use those 4 bits to represent non-negative integers (called *unsigned* data type in conventional programming languages like *C/C++*, *Java*, *Fortran*, etc.), the range of values that can be represented is thus $(0, 1, 2, \dots, 15)$. If positive as well as negative numbers are needed, half of the combinations are used to represent positive and the remaining half represent negative numbers. It is necessary, therefore to use one bit from the set of bits allocated (e.g., typically the *Most Significant Bit* or *MSB* is used, in this case, bit number 3) to represent the sign of a number. There are several different binary number representational conventions for signed and unsigned numbers. The most notable ones are:

1. Sign Magnitude.
2. One's Complement.
3. Two's Complement.

An example of 4-bit signed numbers is presented in the Table 1.1, for the three listed formats.

1.4.2 Sign-Magnitude Format

As depicted in Table 1.1, signed integers (positive and negative values) in this format use the *MSB* bit to represent the sign of the number and the remaining bits are used to represent its magnitude. A 16-bit sign-magnitude formant representation is depicted in Fig. 1.13 below.

This format thus, has two possible representations for 0, one with a positive sign and one with a negative sign as depicted in Table 1.1. This poses additional

Table 1.1 Example of 4-bit number representations

Decimal value	Sign-magnitude	One's-complement	Two's-complement
Binary number representations			
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	1000	1111	-
-1	1001	1111	1111
-2	1010	v1111	1110
-3	1011	1111	1101
-4	1100	1111	1100
-5	1101	1111	1011
-6	1110	1111	1010
-7	1111	1111	1001
-8	-	-	1000

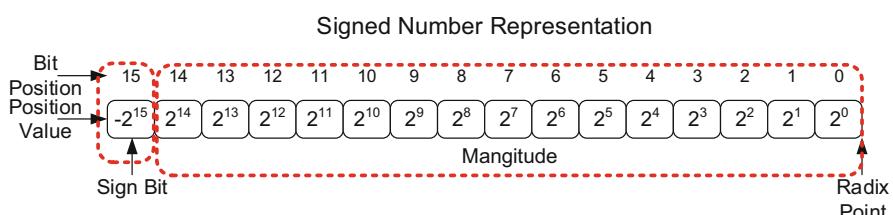


Fig. 1.13 Signed magnitude format

complications in designing the hardware to carry out operations. This issue is discussed further in the following sections. With 4 bits the range of values cover the interval $[-7, +7]$. With 16 bits the range is defined to the following interval $[-32, 767, +32, 767]$. In general, with n bits in a sign-magnitude format, only the integers in the range from $-(2n - 1 - 1)$ to $+(2n - 1 - 1)$ are represented. This format has two drawbacks. The first drawback already mentioned is that it has two different representations for 0. The second drawback is that it requires two different rules one for addition and one for subtraction, and a way to compare magnitudes to determine their relative values prior to applying subtraction. This in turn would require more complex hardware to carry out those rules.

1.4.3 One's-Complement Format

As depicted in Table 1.1, negative values $-x$ is obtained by negating or complementing each bit of the binary representation of a positive integer x . For an $n - bit$ binary representation of a number x , the following is its one's-complement:

$$x \equiv b_{n-1} \dots b_2 b_1 b_0,$$

n —bit representation of a number x

$$x \equiv \bar{b}_{n-1} \dots \bar{b}_2 \bar{b}_1 \bar{b}_0,$$

n —bit representation of one's complement of number x , where \bar{b}_i is a complement of bit b_i . The following holds:

$$x + \bar{x} \equiv 1 \dots 111 = 2^n - 1 \quad (1.1)$$

Similarly to sign-magnitude format, *MSB* is used to represent the sign of a number. A positive number will have an *MSB* value of “0” which after the complement operation will become “1” leading to a negative integer number. The remaining $n - 1$ bits will represent a number itself if positive; otherwise, they will represent one's complement. Applying Eq. 1.1 the following expression depicts one's-complement representation format:

$$\begin{aligned} x_{(\bar{1})} &\triangleq \begin{cases} x, & x \geq 0 \\ |\bar{x}| & x < 0 \end{cases} \\ &= \begin{cases} x, & x \geq 0 \\ (2^n - 1) - |x| & x < 0 \end{cases} \end{aligned} \quad (1.2)$$

$$= \begin{cases} x, & x \geq 0 \\ (2^n - 1) + x & x < 0 \end{cases}$$

Similarly to sign-magnitude representation, with 4 bits we can represent integers in the range defined by the interval $[-7, +7]$, also as depicted in Table 1.1. In general, with n bits, the one's-complement format can represent the integers in the range from $-(2n-1-1)$ to $+(2n-1-1)$. The one's-complement format is superior to sign-magnitude format in that the addition and subtraction require only one rule; specifically, that of the addition, since subtraction can be carried out by performing addition on the one's-complemented number as depicted below by applying Eq. 1.2.

$$z_{(1)} = x_{(\bar{1})} - y_{(\bar{1})} = x_{(\bar{1})} - |\bar{y}_{(\bar{1})}| = x_{(\bar{1})} + 2^n - 1 + y_{(\bar{1})} \quad (1.3)$$

It turns out that the addition of one's-complement numbers is a bit complicated to implement in hardware. Also, an additional extra bit is required to represent the least significant bit (2^0) to manage overflow. This problem is alleviated by the two's-complement representation discussed next.

1.4.4 Two's-Complement Format

The n -bit two's-complement number of a positive integer x is defined by the following expression:

$$\begin{aligned} \tilde{x} &= \bar{x} + 1 = 2^n - x \\ x + \tilde{x} &= 2^n \end{aligned} \quad (1.4)$$

As depicted in Table 1.1, the disadvantage of having two representations for the number zero is eliminated. As before, *MSB* is used to represent the sign of the number. Using Eq. 1.3 the two's-complement format representation is given by:

$$\begin{aligned} x_{\bar{1}} &\hat{=} \begin{cases} x, & x \geq 0 \\ |\tilde{x}| & x < 0 \end{cases} \\ &= \begin{cases} x, & x \geq 0 \\ (2^n) - |x| & x < 0 \end{cases} \\ &= \begin{cases} x, & x \geq 0 \\ (2^n) + x & x < 0 \end{cases} \end{aligned} \quad (1.5)$$

With two's-complement representation, obtained by shifting to the right by incrementing one's-complement representation, the problem of two zeros is alleviated. Consequently, the range of the negative numbers has increased by one compared to previous representations, as depicted in Table 1.1. With 4 bits the range of integer values is defined by the interval $[-8, +7]$, also as depicted in Table 1.1. In general, with n bits, the two's-complement format can represent the integers in the range from $-(2n - 1)$ to $+(2n - 1 - 1)$. The following lists the advantages of the two's-complement format representation:

1. It is compatible with the notion of negation, that is, the complement of complement is the number itself.
2. It unifies the subtraction and addition operations since subtractions are essential additions of two's-complement representation of a number.
3. For a summation of more than two numbers, the internal overflows do not affect the final result so long as the result is within the range; adding two positive numbers results in a positive number, and adding two negative numbers gives a negative result.

Due to these properties, the two's complement is the preferred format in representing negative integer numbers. Consequently, almost all current processors, including DSP's implement signed arithmetic using this format and provide special functions to support it.

1.5 Fix-Point DSP's

The fixed-point DSP's hardware supports only fixed-point data types. Their hardware is thus more restrictive performing basic operations only on fixed-point data types. With software emulation, the fixed-point DSPs can execute floating-point operations. However, floating-point operations are done at the expense of performance due to a lack of floating-point hardware. Lower-end fixed-point DSPs are 16-bit architectures. That is, the processors' word length is 16 bit and its basic operations use 16-bit data types. Typically, 16-bit DSPs also support double-precision— $2 \times 16 = 32$ -bit data types. This extended support may come at the expense of the performance of the processor depending on their hardware architecture and design. There are a number of possible fixed-point representations that DSP hardware may support. One example was presented in Table 1.1.

For 16 bit representations, the ranges of numbers are given in the Table 1.2. Modern devices' architectures support two's-complement integer formats.

The range of values (commonly referred to in the literature as dynamic range) is proportional to the number of bits used to represent a number. If the result of the operation exceeds the precision of the data type, in the worst-case scenario the resulting number will overflow and wrap around generating a large error, or at best if it is handled (by hardware, setting the overflow flag in processors arithmetic and logic unit's status register or by software, checking the input

Table 1.2 Range of values represented by a 16 bit and 32 bit DSP's

<i>Unsigned fixed-point numbers</i>			
16-Bit		32-Bit	
	Sign-magnitude		One's-complement
Min value	$2^0 = 0$		–
Max value	$2^{16} = 65536$		–
	Sign-magnitude		Two's-complement
Min value	$2^0 = 0$	One's-complement	Two's-complement
Max value	$2^{32} = 4294967296$	–	–
	Sign-magnitude		Two's-complement
16-Bit		One's-complement	Two's-complement
Min value	$-2^{16-1} + 1 = -32767$	$-2^{16-1} + 1 = -32767$	$-2^{16-1} + 1 = -32768$
Max value	$+2^{16-1} - 1 = +32767$	$+2^{16-1} - 1 = +32767$	$+2^{16-1} - 1 = +32767$
	Sign-magnitude		One's-complement
32-Bit		One's-complement	Two's-complement
Min value	$-2^{32-1} + 1 = -2147483647$	$-2^{32-1} + 1 = -2147483647$	$-2^{32-1} = -2147483648$
Max value	$+2^{32-1} - 1 = +2147483647$	$+2^{32-1} - 1 = +2147483647$	$2^{32-1} - 1 = +2147483647$

values before the operation to detect potential overflow) it will be saturated to the maximal/minimal value of corresponding data type leading to truncation error. Since most common DSP operations require multiply and accumulate operations, these kinds of representations where the magnitude of the number is directly mapped in the processor require special handling to avoid truncation effects. In addition, exceeding the precision provided by the dynamic range of the data type typically introduces non-linear effects producing large errors and sometimes breaks the algorithm (Fig. 1.14).

1.6 Fixed-Point Representations Based on Radix-Point

One way to view possible fixed-point representations by a processor is based on the implied position of the radix point. In the examples of integer fixed-point representations discussed previously, zero bits were used after the radix point. This implies the following representation depicting the integer formants in DSP:

$$x = \pm(d_n B^n + \dots + d_2 B^2 + d_1 B^1 + d_0 B^0.)$$

Example 1.7 A DSP uses 4 bits to represent input fixed-point all integer numbers. Two's complement format is used for negative numbers. The tables below indicate the resulting numbers if the operations are carried using 4-bits (Table 1.3).

In the cases of overflow, it becomes necessary to handle it to minimize the resulting error. In the table above the resulting overflow, errors are $6 + 5 = 11$

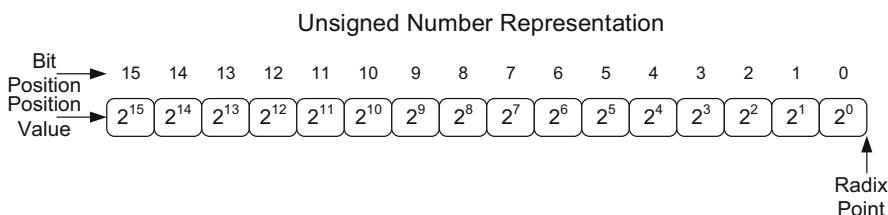


Fig. 1.14 The 16-bit unsigned and signed data type's representation

Table 1.3 Example of 4 bit signed magnitude number operations with 4 bit resulting number

Operand 1 4-bits binary (decimal)	Operation	Operand 2 4-bits binary (decimal)	Resulting number	Comment
0011(+3)	+	0010(+2)	0101(+5)	No overflow
0110(+6)	+	0101(+5)	1011(-5)	Overflow
1101(-3)	+	0111(+7)	0100(+4)	No overflow
1010(-6)	+	1000(-8)	0010(+2)	Overflow

compared to the erroneous result of (-5) , or $-6 - 8 = -14$ vs. $+2$; both being 16. Overflow can be avoided by doubling the precision of the resulting number; that is using 8 bits as depicted in Table 1.4.

The next table depicts the results of the multiplication of two 4-bit numbers with a 4-bit resulting number. The errors due to overflow are large even when the 4 most significant bits (MSB) are used for resulting numbers. For example $(+7)x(+6) = (+42)$ but the resulting number from the 4 MSB's is 2 which introduces an error of 40. Similarly, $(-6)x(-5) = (-35)$ with the resulting number (-2) and error of 33 (Table 1.5).

Similarly to addition operation, the overflow can be avoided for multiplication if the resulting precision is doubled compared to input operands. The table below demonstrates this case (Table 1.6).

In all but trivial algorithms is possible to keep advancing the precision of the resulting number. Those algorithms require a number of iterations involving intermediate data from input data to generate the resulting output. To achieve error-free operation each intermediate output has to have a double number of bits compared to its inputs. Iterative application of this approach will quickly exceed the hardware capabilities of the DSP (e.g., 32-bit or 64 bit int data type). However, several techniques enable DSP developers to bind the errors to within the tolerated margins as established by the application. This issue will be discussed further in the next chapter.

Table 1.4 Example of 4 bit signed magnitude number operations with 8 bit resulting number

Operand 1 4-bits binary (decimal)	Operation	Operand 2 4-bits binary (decimal)	Resulting number 8-bits binary (decimal)
0011(+3)	+	0010(+2)	0000 0101 (+5)
0110(+6)	+	0101(+5)	0000 1011 (+11)
1101(-3)	+	0111(+7)	0000 0100 (+4)
1010(-6)	+	1000(-8)	0000 0010 (-12)

Table 1.5 Example of 4 bit signed magnitude number operations with 4 bit resulting number

Operand 1 4-bits binary (Decimal)	Operation	Operand 2 4-bits binary (Decimal)	Resulting number	Comment
0010(+2)	×	0011(+3)	0110(+6)	No overflow
0111(+7)	×	0110(+6)	0010(+2)	Overflow
1101(-3)	×	0010(+2)	0110(+6)	No overflow
1010(-6)	×	1011(-5)	1111(-2)	Overflow

Table 1.6 Example of 4 bit signed magnitude number operations with 8 bit resulting number

Operand 1 4-bits binary (decimal)	Operation	Operand 2 4-bits binary (decimal)	Resulting number 8-bits binary (decimal)
0010(+2)	×	0011(+3)	0000 0110 (+6)
0110(+7)	×	0110(+6)	0010 1010 (+42)
1101(-3)	×	0110(+2)	0000 0110 (-6)
1010(-6)	×	1011(+5)	1110 0010 (-30)

There are alternative representations that have better properties than the common integer formats presented so far. One such representation requires all numbers to be scaled within the interval $[-1, 1)$. This is all fractional representation using fixed-point architecture. Note, this representation is not to be confused with fractional numbers in floating-point representations which use different formats and rules in the hardware to perform floating-point operations.

In all fractional representations, allocated bits are used to cover the fixed dynamic range between -1 and 1 . The larger the number of bits used for fractional numbers, the finer the representation (finer granularity). This stands in contrast to the previous magnitude representation where the granularity is fixed and is equal to 1 —a constant difference of any two consecutive numbers. Imposing a constant range may potentially be considered a drawback of this fractional representation since it may require keeping track of the scaling factor used to translate the original range of values to a fixed $[-1, 1)$ range. On the other hand, this representation provides much better properties in terms of truncation error as well as overflow.

Truncation error and overflow require special consideration in fixed point integer representation discussed earlier. On the other hand, the 16-bit fractional representations do not require overflow handling in multiplication; multiplication of two fractional numbers with values between -1 and 1 will produce a resulting number also in the same range. The only consideration with the 16-bit fractional representation is underflow which typically does not require special handling. Underflow incurs the error when the result of an operation is less than the granularity of the representation. If this error cannot be tolerated additional rescaling of the intermediate data is required, otherwise the effect of error falls below the granularity of the representation; i.e., smallest representable number.

A fractional fixed-point representation assumes radix-point to be in the left-most position implying that all bits have positional values less than one. The general notation of this fractional representation is:

$$x = \pm \cdot (d_{-1}B^{-1} + d_{-2}B^{-2} + \dots + d_{-m}B^{-m})$$

Presented integer and fractional fixed-point representations depict two possible number representational schemes utilizing two extreme positions of implied radix-point. Since radix-point position defines the notation, a formal definition of such representational scheme is based precisely on it. Let N be the total number of bits used to represent a number. Also let p denote the number of bits to the left of the radix point specifying the integral portion of a number, and with q number of bits to the right of radix-point specifying the fractional portion of a number. Notation $\mathbb{Q}_{p,q}$ specifies the format of the representation used as well as the position the implied radix-point as well as the precision of the representation. For example, the unsigned integer fixed-point format is expressed as $\mathbb{Q}_{16,0}$ since all bits lay to the left of radix-point. Consequently, signed 16-bit integer fixed-point format is denoted by $\mathbb{Q}_{15,0}$ with 1 bit used to represent the sign of a number. The all fractional representation uses $\mathbb{Q}_{0,16}$ and $\mathbb{Q}_{0,15}$ format for unsigned and signed numbers respectively. In general, for unsigned numbers the relationship between total number of bits N and p, q is:

$$N = p + q \quad - \text{unsigned}$$

For signed numbers, the following relationship holds:

$$N = p + q + 1 \quad - \text{signed}$$

In light of introduced notation, a number using $\mathbb{Q}_{p,q}$ as a binary signed format has a value that can be computed by the following expression:

$$\begin{aligned} \text{Num} &= \left(-b_{N-1}2^{N-1} + b_{N-2}2^{N-2} + b_{N-3}2^{N-3} + \dots + b_12^1 + b_02^0 \right) 2^{-q} \\ &= -b_{N-1}2^p + \sum_{k=0}^{N-2} b_k 2^{k-q} \end{aligned}$$

For unsigned numbers the following expression can be used:

$$\begin{aligned} \text{Num} &= \left(b_{N-1}2^{N-1} + b_{N-2}2^{N-2} + b_{N-3}2^{N-3} + \dots + b_12^1 + b_02^0 \right) 2^{-q} \\ &= \sum_{k=0}^{N-1} b_k 2^{k-q} \end{aligned}$$

1.7 Dynamic Range

Now a formal definition of the dynamic range of a data representation can be stated. The dynamic, range given in the *dB* scale, is defined as the ratio of the largest number (*Max*) and the smallest positive number greater than zero (*Min*) of a data representation. It is computed by the following expression:

$$DR = 20 \log \left(\frac{Max}{Min} \right)$$

Dynamic Range of signed and unsigned integer and fractional 16-bit representations are given in Table 1.7.

1.8 Precision

Earlier, we have introduced the concept of the granularity of a representation. Here, it is formally defined as the precision of a representation. The precision of a representation is thus the difference of two consecutive numbers. It has to be noted

Table 1.7 Dynamic range and precision of 16-bit signed and unsigned integer and fractional representations

(16-bits)	Dynamic range	Dynamic range (dB)	Precision
Unsigned integer	[0,65535]	$20 \log \left(\frac{2^{16}-1}{2^0} \right) > 96 \text{ dB}$	1
Signed integer	[-32768,32767]	$20 \log \left(\frac{2^{15}-1}{2^0} \right) > 90 \text{ dB}$	1
Unsigned fractional	[0,0.9999847412109375]]	$20 \log \left(\frac{1-2^{-16}}{2^{-16}} \right) > 96 \text{ dB}$	2–16
Signed fractional	[-1,0.9999847412109375]]	$20 \log \left(\frac{1-2^{-15}}{2^{-15}} \right) > 90 \text{ dB}$	2–15

that this difference is the smallest step between any two representations. In the Table below, largest and smallest positive and negative values as well as corresponding precessions for fractional and integer 16-bit data types are given.

In addition to 16-bit signed and unsigned representations already discussed; namely $\mathbb{Q}_{16,0}$, $\mathbb{Q}_{15,0}$ integer and $\mathbb{Q}_{0,16}$ and $\mathbb{Q}_{0,15}$ fractional, there is a whole range of representations in between that could be used that combine pure integer and pure fractional representations.

Different $\mathbb{Q}_{p,q}$ formats are depicted in the next Fig. 1.15. The following table uses format definitions in the figure to present the ranges of possible 16-bits signed numbers that can be represented in a DSP.

Table 1.8 below summarizes the properties of all possible 16-bit representations in terms of a number of integer bits (p), a number of fractional bits (q), largest positive value, least negative value, and precision for various \mathbb{Q} formats (Table 1.9).

1.9 Background Information

One of the most important century in terms of scientific thought is the eighteenth century AD. Specifically in mathematics, the Euler formula (sometimes also called the Euler identity), states:

$$e^{ix} = \cos(x) + i \sin(x)$$

where i denotes the imaginary constant, i.e., $i = \sqrt{-1}$. The special case of the formula with $x = \pi$ gives the identity

$$e^{i\pi} + 1 = 0$$

This equation connects the fundamental numbers i , π , e , 1, and 0 (zero). It gives fundamental operators for $+$, \times , exponentiation, and the important relation $=$. It also links into the problem of the root of unity, that is:

$$z^n = 1$$

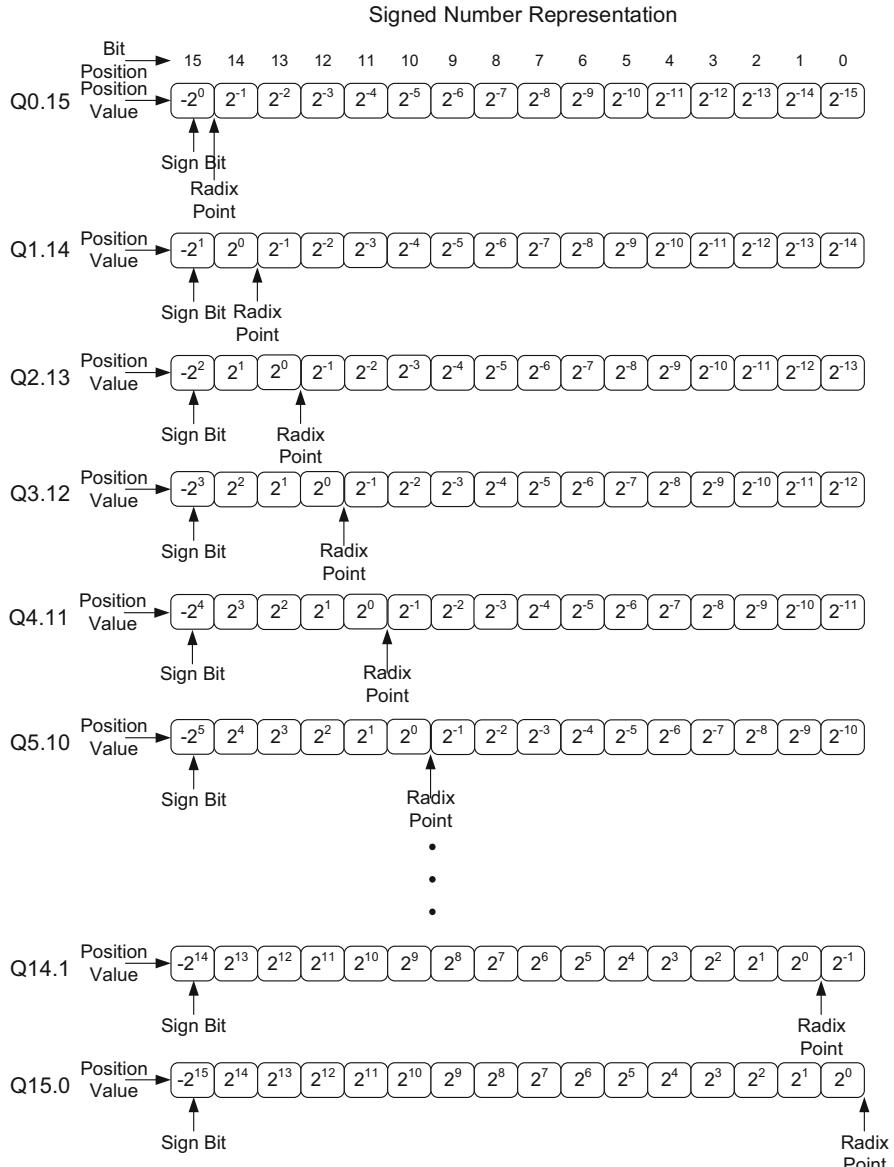


Fig. 1.15 Possible $\mathbb{Q}_{p,q}$ format representations of 16-bit data length

The solution of roots of unity, which is any complex number that yields 1 when raised to some positive integer power n . Noting that from theory of mathematics, a polynomial of degree n with real, or complex, coefficients has n roots.

Table 1.8 Maximal, minimal, and precision values for integer and fractional 16-bit fixed-point representations

<i>Unsigned fractional and integer 16-bit representations</i>					
Format	Number of integer bits	Number of fractional bits	Largest value	Smallest value	Precision
Q0.16	0	16	0.9999847412109375	0.0	0.0000152587890625
Q16.0	16	0	65535.0	0.0	1.0
<i>Signed fractional and integer 16-bit representations</i>					
Format	Number of integer bits	Number of fractional bits	Largest positive value	Smallest negative value	Precision
Q0.15	0	0	0.9999847412109375	-1.0	0.000030517578125
Q15.0	15	0	32767.0	-32768.0	1.0

Table 1.9 Maximal, minimal, and precision values for integer and fractional 16-bit signed fixed-point representations

Format	Number of integer bits	Number of fractional bits	Largest positive value in decimal ($0 \times 7FFF$)	Smallest negative in decimal value (0×800)	Precision
<i>Signed 16-bit representations</i>					
Q0.15	0	15	0.999969482421875	-1.0	0.000030517578125
Q1.14	1	14	1.99993896484375	-2.0	0.00006103515625
Q2.13	2	13	3.9998779296875	-4.0	0.0001220703125
Q3.12	3	12	7.99951171875	-8.0	0.000244140625
Q4.11	4	11	15.99951171875	-16.0	0.00048828125
Q5.10	5	10	31.9990234375	-32.0	0.0009765625
Q6.9	6	9	63.998046875	-64.0	0.001953125
Q7.8	7	8	127.99609375	-128.0	0.00390625
Q8.7	8	7	255.9921875	-256.0	0.0078125
Q9.6	9	6	511.984375	-512.0	0.015625
Q10.5	10	5	1023.96875	-1024.0	0.03125
Q11.4	11	4	2047.9375	-2048.0	0.0625
Q12.3	12	3	4095.875	-4096.0	0.125
Q13.2	13	2	8191.75	-8192.0	0.25
Q14.1	14	1	16383.5	-16384.0	0.5
Q15.0	15	0	32767.0	-32768.0	1.0

Authors are encouraged to read more used references in this chapter such as Oppenheim et al. (1999), Ingle and Proakis (2007), Zölzer (1998), Knuth (1997), Wiki (2021), Kondoz (2004), Quartieri (2002), Kuo and Gan (2005), and VisualDSP++.

1.10 Exercises

Exercise 1 Describe the pro's and con's of fixed-point signed data representation for 8 bit ("char" in C) representation in terms of minim, maximum value, operations such as "+" and "-", etc.

Exercise 2 What are benefits of Qm.n representation over fixed point signed data representation of equal length (e.g., same number of bits).

Exercise 3 Use Euler's formula for the n th roots of unity to provide a general solution to the $z^n = 1$ noting that polynomial of order n for $n > 0$ has n roots?

References

- [Jayant1984] Jayant & Noll, "Digital Coding of Waveforms-Principles and Applications to Speech and Video", Chapter 3, Sampling and Reconstruction of Bandlimited Waveforms, Prentice Hall, 1984
- [Oppenheim1999] Oppenheim, Schafer & Buck, "Discrete-Time Signal Processing", Chapter 6, Overview of Finite -Precision Numerical Effects, Prentice-Hall, 1999
- [IngleProakis2007] Ingle & Proakis, "Digital Signal Processing using MATLAB", Chapter 9, Finite Word-Length Effects, Thomson, 2007
- [Zolzer1998] Udo Zölzer, "Digital Audio Signal Processing, Chapter 2, Quantization, Wiley, 1998
- [ConwayGuy1999] J. H. Conway, R. K. Guy, "The Book of Numbers", Copernicus an imprint of Springer Verlag, New York, 1999
- [Knuth1997] D. Knuth, "The Art of Computer Programming", Volume 2, Seminumerical Algorithms, Third Edition, Addison-Wesley, 1997
- [Wiki2021] http://en.wikipedia.org/wiki/Real_number
- [Kondoz2004] Kondoz, "Digital Speech, Coding for Low Bit Rate Communication Systems, Wiley, 2004
- [Jensen2005] Jensen, Batina, Hendriks & Heusdens, "A Study of the Distribution of Time-Domain Speech Samples and Discrete Fourier Coefficients", Proceedings of SPS-DARTS, 2005
- [Quartieri2002] Quartieri, "Discrete-Time Speech Signal Processing: Principles and Practice, Prentice-Hall, 2002
- [KuoGan2005] Kuo & Gan, "Digital Signal Processors-Architectures, Implementations, and Applications", Chapter 3, Implementation Considerations, Prentice-Hall, 2005
- [VisualDSP++] VisualDSP++ C/C++ Compiler and Library Manual for Blackfin Processors, Analog Devices, Inc. One Technology Way Norwood, Mass. 02062-9106, www.analog.com

Chapter 2

Signal Processing Background



Overview

A signal carries information. In this chapter, we provide an overview of various signals and how the information of the signal is retained and used for different applications. The information can be contaminated by some other types of signals. Such contaminating signal is normally termed as noisy signal. The signal is processed, analyzed, filtered and enhanced. The enhanced signal can be used for applications as intended. The analysis can be stochastic or deterministic. Most real world signals are random and unpredictable. The deterministic analysis is only partially fruitful for common real world signals since in this analysis the information and signal details must be known. Unpredictable signals are stochastic processes. These are processed and analyzed by applying statistical methods. The statistical analysis is more complex than the deterministic one because the information in the signal is varying at random and has to be processed applying criteria of stochastic process. In this chapter, we have an overview of different analysis techniques.

2.1 Basic Concepts

The basic concepts discussed here are classified in three main topics:

1. Signal and Transformation

Overall, the book deals with signals. Signals are functions returning as values certain objects from a given domain. Signals are analyzed in the time domain or in the frequency domain. In general, the domain is not restricted. For our purpose it is commonly sufficient to work with real or binary numbers. Signals do not occur in isolation but rather they are occurring in complex sets. These sets are mostly partially ordered and together they form a signal process. In such a signal process the sequences of signals can contain gaps. Weights can be specified for showing the importance of individual signals. The ordering of

the signal is processed through some transformations. We provide a summary of such transformations. Signals can be periodic, non-periodic, or random. Most commonly analyzed real world signals are random.

2. Spectral Analysis

Signal power and frequency information are observed in signal spectrum. Common analysis can be parametric or non-parametric. Fast Fourier Transform (FFT) is a common analysis tool for signals that are mainly stationary. However, most real world signals are non-stationary and random. Spectral analysis can be temporal, spatial, or a combination of both. A common spatial and temporal spectral analysis approach is windowing approach. A random signal is often analyzed as a quasi-stationary signal by applying windowing, and then a Fourier transform is computed. This is also called [Short Time Fourier Transform](#) (STFT). In addition, these signals are processed applying auto-correlation, mean squared error, covariance function, and power spectral density. Multivariate spectral analysis, e.g. a filterbank can have different sub-sampling rates thus forming sub-components. Each sub-component is known as a sub-band. This is known as signal decomposition.

The signal is decomposed into different frequency bands by frequency selective filtering for analysis and synthesis. Here we have a brief introduction about the tree structured and quadrature mirror filter banks and their applications.

3. Adaptive Filtering

Most signals are mixed with some other signals. An undesired signal is considered to be noise. Adaptive filters are commonly used to process noisy signals in order to obtain desired information. Standard [Finite Impulse Response digital filter](#) (FIR) or [Infinite Impulse Response digital filter](#) (IIR) digital filters can have a large number of filter coefficients and the computational cost involved in applying them can be high. The filters are used to process the signal in such a way that the parameters of the signal are updated and adjusted iteratively so that there is a relation between the input and output signal. The filters are structured in such a way that the output of the signal can closely replicate the desired input, also referred to as observation. An adaptive filter is applied on the basis of optimization criteria, algorithm selection, filter structure and the type of the signal. The optimization of the error function is a common optimization criterion. The optimized error criterion can be mean squared error, or least squared error, or weighted least squared error, or some other criterion. The optimization criterion can be based on techniques such as the normal equations, Newton method, Gram Schmidt method, or Cholesky method. Based on the employed algorithm and optimization criteria, the filter parameters are selected and optimized iteratively. The filter structures can be linear, non-linear, lattice or canonical. The optimization of the parameters also depends on the selected type of filters and number of parameters selection. The input digital signal can be fed to the filter as one sample at a time or a collection of samples of the signal covering a specific interval termed frame. Examples of adaptive techniques are: the Wiener filter, the Kalman filter, the particle filter, adaptive array processing, beam forming. Some adaptive tools are Independent Component Analysis (ICA),

and Principle Component Analysis (PCA). We will discuss some of the adaptive techniques and their application in this section and in Part III.

2.2 Signals and Information

Signals convey information. This can be in different types and formats: speech, images, video, temperature, electrocardiogram, electromyogram, seismic etc. The signals are processed, and analyzed on the computer for their internal or external representational application. The representations can be discrete, continuous, or a mixture of these two. A typical signal processing objective is to remove redundancies, and extract the meaningful information for intelligent systems. These are achieved applying techniques from multiple areas such as combinations of signal processing, control system, Machine Learning (ML), Artificial Intelligence (AI), etc. The extracted meaningful information often is used to recognize interesting patterns, identify significant events, and detect targets. Such systems are often intelligibly automated by ML techniques, and AI algorithms. The signal changes often unpredictably from time to time, from subject to subject, events to events, environmental or subjective situations.

2.3 Signal Processing

Signal processing is in fact used in almost all areas; the areas such as space science, medical science, telecommunication, in defence technology and military applications, industrial science, oil and gas industry to name a few. The signal processing in general transforms the signal. The type of conversion varies from one area to other areas and from one application to another application. We have mentioned some application areas. For example, the perception of the speech signal is transformed semantically into spoken words by the function of the brain. Similarly we capture images and movements of a person in order to process this via video analysis. A signal processing application area is forensic medicine where the fingerprint is used for identifying a person. Other areas include space science, sensors, tracking, monitoring, and detection, biomedical signal processing medical image analysis, diagnosis, data mining, process monitoring, radar signal processing, and intelligent traffic control system. All those processes can be described as stochastic processes and can also be formulated using Brownian motion.

Two most common signal types are deterministic and random, or non-deterministic. The deterministic signals can be modeled by a mathematical formula and hence considered deterministic. An example of a deterministic signal is the sinusoidal signal, which has the same repetitive pattern, known characteristics, and predictability. Non-deterministic signal, on the other hand, cannot be described by a mathematical function. The behavior of non-deterministic or random signals

change over time. Speech signals are typical non-deterministic signals. They are analyzed as discrete time sequences in small time chunks or frequency intervals as quasi-periodic signals. The duration of intervals used to process the signals depends on the type of signals and its characteristics. The characteristics of each interval are assumed to remain the same or at least similar. They are analyzed as ensemble signals.

In Part III, we discuss about speech signals, some biomedical signals such as **Electrocardiogram** (ECG) and **Electromyography** (EMG), multimedia signal and seismic signal. Some biomedical signal e.g. electrocardiogram (ECG) have periodic patterns.

2.4 Discrete Signal Representations

A signal is generally processed in discrete forms. One does not have influence on the original form, which is determined by the source. However, one may want to transform the input from a continuous into a discrete form in order to process the signals. There are several reasons for it:

- Counting signals
- Computation of mean or variance of the signal

The signal processing starts with the discretization of the signal from a continuous signal as shown in Fig. 2.1. A continuous pressure waveform $s(t)$ is captured by a microphone which is afterwards amplified, anti-aliased via a low pass filter following a so-called sample and hold process to be transformed into the discrete-time signal $s[n]$.

When the signal is decomposed into discrete representations, these are called impulses. These impulses are analyzed as impulse delta function, step function, or saw-tooth wave function. We will introduce some common analytical signals next.

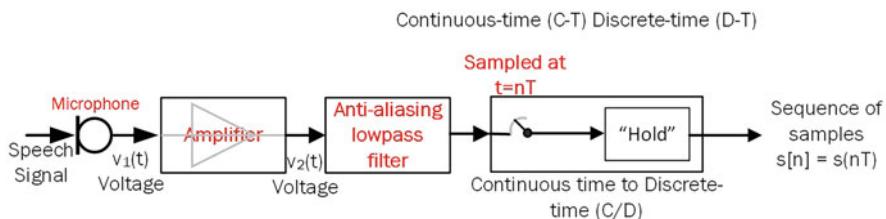


Fig. 2.1 Signal is processed from continuous to discrete representation

2.5 Delta and Impulse Function

The delta signal is approximated by a rectangular pulse of narrow width and the long length such that the area covered under the rectangular signal is 1. The delta function is shown in figures as having a segment at non-zero values with length $1/l$ and height l such that the area is 1. The delta function is also known as generalized function or distribution. The delta function is also commonly used in the discrete system as an impulse function. The impulse aka dirac¹ delta function is generally used as a common tool to define systems in continuous or discrete time. The characteristic property of an impulse function in point $t = 0$ is: at $t = 0$, the dirac delta function is assumed to have infinite value, and it is zero elsewhere, with unit area under the curve.

The integral in Eq. 2.1 is true for intervals $a + \epsilon < 0 < a - \epsilon$ where $p(t - a)$ is an impulse function located at $t = a$ as shown in Fig. 2.2.

$$\delta(t - a) \text{ such that } \begin{cases} p(t - a) &= 0, \quad t \neq a \\ \int_{a-\epsilon}^{a+\epsilon} p(t - a) dt &= 1, \quad \epsilon > 0 \end{cases} \quad (2.1)$$

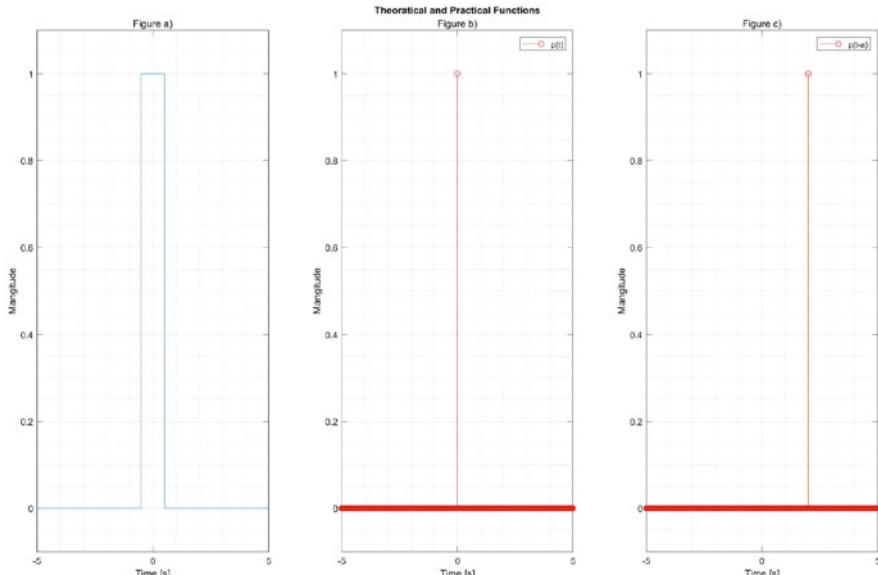


Fig. 2.2 Delta function

¹ This term is used in honor of the Paul Adrien Maurice Dirac (1902–1984), who was an English theoretical physicist who made fundamental contributions to the early development of both quantum mechanics and quantum electrodynamics. He held the Lucasian Chair of Mathematics at the University of Cambridge and spent the last fourteen years of his life at Florida State University.

$$\int_{-\infty}^{+\infty} f(t)p(t-a)dt = \int_{a-\epsilon}^{a+\epsilon} f(t)p(t-a)dt = f(a), \quad \epsilon > 0 \quad (2.2)$$

In discrete domain, the delta function is defined as

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \forall n \neq 0 \end{cases} \quad (2.3)$$

Question 2.1 Apply the dirac/delta integral at $\delta(x + 2)$ in Eq. 2.4.

$$f(x) = x^3 - 3x^2 + 2x - 1 \quad (2.4)$$

Answer: Applying Eq. 2.2 in 2.4, we find

$$\int_{-3}^1 f(x)\delta(x+2)dx = \int_{-3}^1 (x^3 - 3x^2 + 2x - 1)\delta(x+2)dx = f(-2) \quad (2.5)$$

This was obtained by applying the sifting theorem depicted in the Eq. 2.2. Substituting that value of $f(-2)$ in Eq. 2.4 we get

$$f(-2) = (-2)^3 - 3(-2)^2 + 2(-2) - 1 = -25$$

Question 2.2 Apply Dirac/delta integral at $\delta(x - 2)$ in Eq. 2.6.

$$f(x) = \exp(|x| + 3) \quad (2.6)$$

Answer:

$$\int_{-1}^1 f(x)\delta(x-2)dx = \int_{-1}^1 \exp(|x| + 3)\delta(x-2)dx = f(2) \quad (2.7)$$

Substituting, we find

$$f(2) = e^{|2|+3} = e^{|2|+3} = e^5$$

Note that the limits of integration does not include $x = 2$.

Question 2.3 Apply Dirac/delta integral at $\delta(x - \pi)$ in Eq. 2.8

$$f(x) = \cos(3x) + 2 \quad (2.8)$$

Answer:

$$\int_0^{\infty} f(x)\delta(x-\pi)dx = \int_0^{\infty} [\cos(3x) + 2]\delta(x-\pi)dx = f(\pi) \quad (2.9)$$

Substituting the value for $x = \pi$ in the above equation we get:

$$f(\pi) = [\cos(3\pi) + 2] = -1 + 2 = 1$$

as

$$\cos(3\pi) = -1$$

The following question involves techniques introduced in Chap. 3.

Question 2.4 Determine impulse response of the difference equation of a system in the following function:

$$y[n] = 0.3x[n] + 0.45x[n - 1] + 0.6x[n - 2] \quad (2.10)$$

Answer: Taking the z-transform of Eq. 2.10 we arrive at Eq. 2.11:

$$Y(z) = 0.3X(z) + 0.45X(z)z^{-1} + 0.6X(z)z^{-2} \quad (2.11)$$

From Eq. 2.11 we find the transfer function $H(z)$ in Eq. 2.12

$$H(z) = \frac{Y(z)}{X(z)} = 0.3 + 0.45z^{-1} + 0.6z^{-2} \quad (2.12)$$

Taking the inverse transform of the Eq. 2.12 gives the impulse response of the system in Eq. 2.13 by applying the Eq. 2.3 where $\delta[n]$ is the impulse/delta signal in discrete time, we obtain

$$h[n] = 0.3\delta[n] + 0.45\delta[n - 1] + 0.6\delta[n - 2] \quad (2.13)$$

2.6 Parseval's Theorem

Total energy carried by the signal can be calculated in time domain as well as in the frequency domain. In Eq. 2.14 containing Parseval's Theorem, E_s is energy of the signal s , $f(t)$ is the signal in the time domain and $F(\omega)$ is the frequency domain signal where ω denotes the frequency at some point in frequency domain and t indicates the time. When the signal $f(t)$ is transformed into $F(\omega)$, the integral of the transformation converges to an energy signal which is absolutely integrable.

$$E_s = \int_{-\alpha}^{+\alpha} |f(t)|^2 dt = \int_{-\alpha}^{+\alpha} |F(\omega)|^2 d\omega \quad (2.14)$$

An example of an absolute integrable signal is the rectangular pulse which has a Fourier transformation and is considered an energy signal. On the other hand, the impulse signal, step functions, have Fourier transforms but they are not energy signals as they are not absolutely integrable signals. The square amplitude spectrum $|F(\omega)|^2$ of some signal $f(t)$ is the energy spectral density because it describes how the signal energy is distributed among the frequencies in the spectrum. Equation 2.15 shows the energy of the signal $f(t)$ distributed in the range between ω_1 and ω_2 .

$$E_\omega = \frac{1}{\pi} \int_{\omega_1}^{\omega_2} |F(\omega)|^2 d\omega \quad (2.15)$$

Question 2.5 Suppose $f(x) = x$ if $0 \leq x \leq 2\pi$. Apply Parseval's theorem to state $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$. Calculate the Fourier coefficients for the function $f(x) = x$ defined above, using integration by parts for $n \neq 0$.

Solution: From Fourier series equation we get:

$$\begin{aligned} c_n &= \frac{1}{2\pi} \int_{-\pi}^{+\pi} f(x) e^{inx} dx = \frac{1}{2\pi} \int_{-\pi}^{+\pi} x e^{-inx} dx = \\ &= \frac{1}{2\pi} \left[x \frac{e^{-inx}}{in} \right]_{-\pi}^{+\pi} - \frac{1}{2\pi} \int_{-\pi}^{+\pi} \frac{xe^{-inx}}{-in} dx = \\ &= \frac{1}{2\pi} \left[2\pi \frac{(-1)^n}{-in} \right]_{-\pi}^{+\pi} - \frac{1}{2\pi} \left[\frac{e^{-inx}}{-n^2} \right]_{-\pi}^{+\pi} = i \frac{(-1)^n}{n} \end{aligned}$$

The c_0 is derived as

$$c_0 = \frac{1}{2\pi} \int_{-\pi}^{+\pi} f(x) dx = \frac{1}{2\pi} \int_{-\pi}^{+\pi} x dx = 0$$

Parseval's identity says that:

$$\frac{1}{2\pi} \int_{-\pi}^{+\pi} f(x) dx = \sum_{-\infty}^{+\infty} |c_n|^2$$

Computing the right side, we get

$$\sum_{-\infty}^{+\infty} |c_n|^2 = \sum_{n \neq 0} |i \frac{(-1)^n}{n}|^2 = 2 \sum_{1}^{+\infty} \frac{1}{n^2}$$

Computing the left side, we get

$$\frac{1}{2\pi} \int_{-\pi}^{+\pi} |x|^2 dx = \frac{1}{2\pi} \left[\frac{x^3}{3} \right]_{-\pi}^{+\pi} = \frac{1}{2\pi} \frac{2\pi^3}{3} = \frac{\pi^2}{3}$$

Putting it all together, we obtain the following:

$$\frac{\pi^2}{3} = 2 \sum_{n=1}^{\infty} \frac{1}{n^2}$$

Interchanging the order, we get:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

2.7 Gibbs Phenomenon

Sinusoidal components that occur at multiples of the fundamental frequency are called harmonics. A continuous periodic signal is approximated with a sufficiently large number of harmonics. Such representations generate ripples in the frequency domain. This is called a Gibbs phenomena. The following example in Fig. 2.3 illustrates this phenomenon. There a continuous rectangular waveform is approximated using 40 sine waveforms. The crests of the sinusoidal waveforms are some deviations of the original form of the rectangular waveform.

In the Fig. 2.4, the jump continuity of the signal is observed for $n = 1, 3, 7, 19, 49$, and 70. Here the rectangular waveform is approximated by the Fourier transform of $4 \sin\left(\frac{mn\pi}{2}\right)$ where $m = 1, 3, 7, 19, 49, 70$.

The Gibbs' phenomenon occurs near a discontinuity in the signal. No matter how many terms are included in the Fourier series representation, there will always be an error in the form of overshoot near the discontinuity. The overshoot will always be about 9% of the size of the jump.

2.8 Wold Decomposition

The Wold theorem states that a zero mean co-variance stationary process z_i can be decomposed into:

1. A stochastic component which is a linear combination of lags of white noise process.

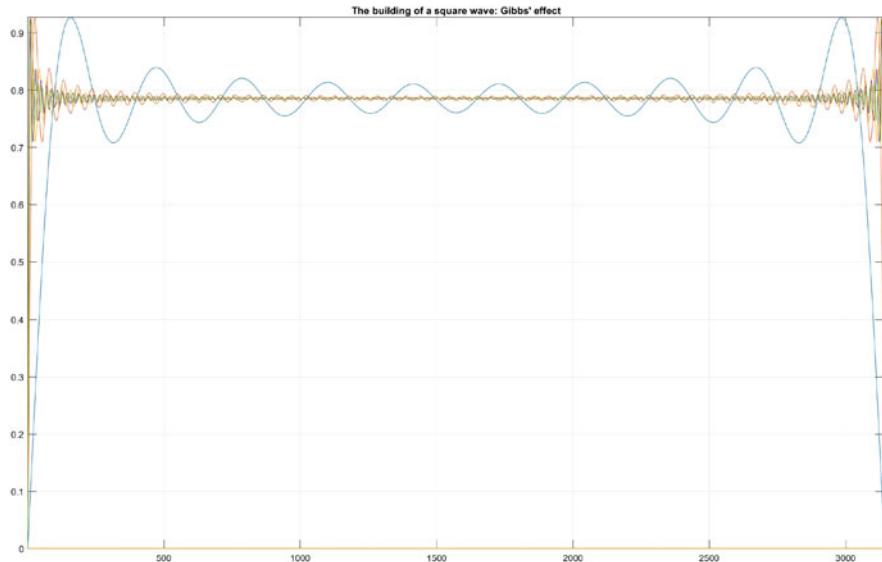


Fig. 2.3 Rectangular waveform and Gibbs phenomenon applying successive approximations 40 sine waveforms

2. A deterministic component, which is uncorrelated with the stochastic component.

$$z_i = \sum_{j=0}^{\infty} \varphi_j x_{i-j} + u_i \quad (2.16)$$

In Eq. 2.16:

1. $\varphi_0 = 1$ and $\sum_{j=0}^{\infty} \varphi_j^2 < \infty$
2. x_i is a white noise which co-variance is σ^2 and $\sigma^2 > 0$
3. Each x_i 's is each unique.
4. The co-variance of x_m and v_n is 0 $\forall m$ and n where $m, n \in \mathbb{Z}$
5. u_i is a deterministic process which is predicted from a linear combination of lagged x .

The Wold decomposition says that any co-variance stationary process has a linear representation. In Eq. 2.16, the x_i is a stochastic or deterministic component and u_i is only deterministic. If $u_i = 0$ then the process z_i is purely non deterministic or stochastic and the z_i can be represented as a moving average (MA) process. The x_i is called an error and it is projected on lagged z_i and lagged x . Therefore, x is uniquely determined and it is orthogonal to lagged z and lagged x . The error x is the residual from the projections and it is an approximated error of z_i . The x_i is white noise and it is not necessary independent and identically distributed (iid) process. The **Auto-Regressive Moving Average** (ARMA) is a type of linear

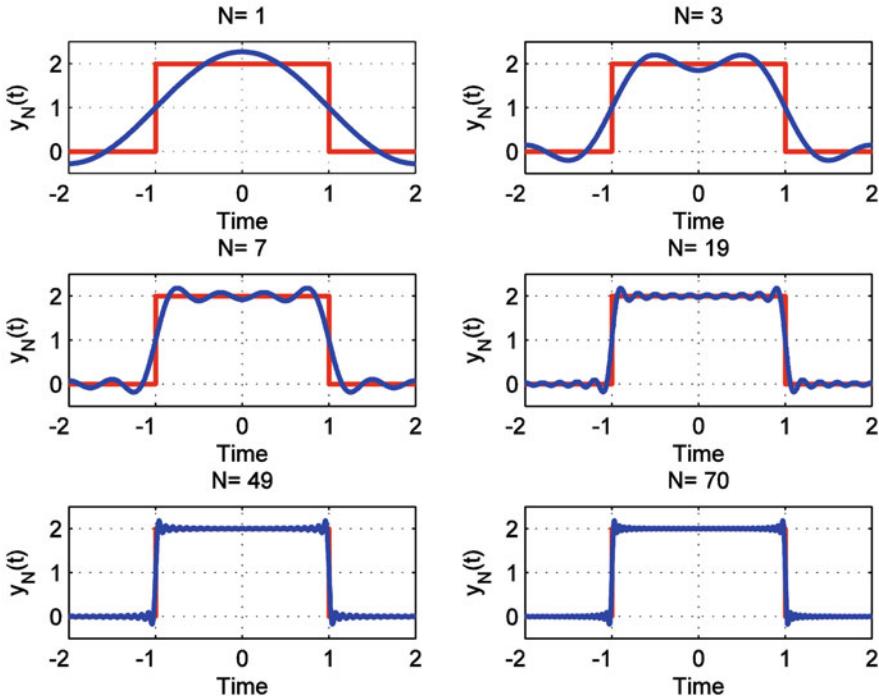


Fig. 2.4 Rectangular waveform and Gibbs' phenomenon approximating $n = 1, 3, 7, 19, 49$ and 70 sine wave-forms

time series representation of a general co-variance stationary or nonlinear process. Applications of the Wold decomposition are given in later parts of this book.

2.9 State Space Signal Processing

A state space model captures the dynamics of the signal behaviors in terms of derivatives of the system states. States are properties such as speed, temperature, frequency, and pressure, which change with time. This type of modeling can be linear or non-linear. Further the input-output relation can be a single input-output model such as SISO and multiple signal input and multiple output such as MIMO. In state space modeling, the signals are analyzed through variables and the state space approach solution is based those variables. Suppose a discrete time system is represented by using N many delays where each loop has at least a delay. The system in the state space representation may include adders and multipliers. In Fig. 2.5, we see the state space representations of a system in different forms

such as continuous-time or discrete time. The continuous-time is generally modeled by differential equations describing the system; the discrete-time representation is modelled with difference equations. The continuous-time system representation uses Laplace transformations, and the discrete-time uses a z-transformation. In Fig. 2.5, L indicates a Laplace transformation in section (c) and in (d) section Z represents a z-transformation. A system with one input and one output is called a **Single Input and Single Output System** (SISO) and a system with multi-inputs and multi-outputs is called a multiple inputs and multiple outputs (MIMO) system. For example, Fig. 2.6 is a SISO system shown left and a MIMO system shown right.

Question 2.6 The input-output relation for a discrete-time system is:

$$y[n+3] + 2y[n+2] + 5y[n+1] = u[n] \quad (2.17)$$

where $u[n]$ is the input and $y[n]$ is the output. Write the discrete-time state space equation for this system.

Solution: Here the input and output are represented by the state variables. Thus the discrete state variables are:

$$x_1[n] = y[n] \quad x_2[n] = y[n+1] \quad x_3[n] = y[n+2]$$

Similarly,

$$x_3[n+1] = y[n+3]$$

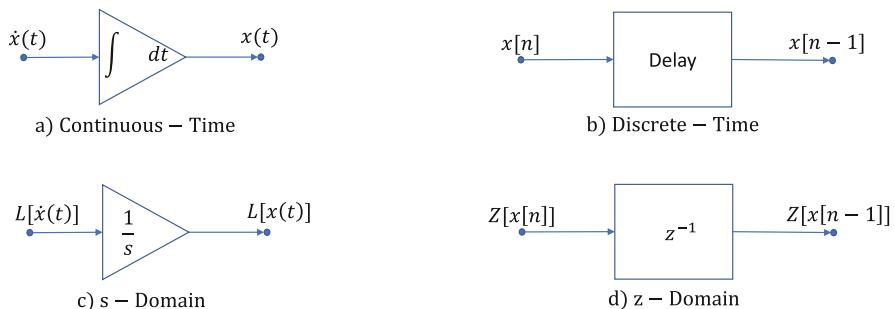


Fig. 2.5 Continuous and discrete SISO system

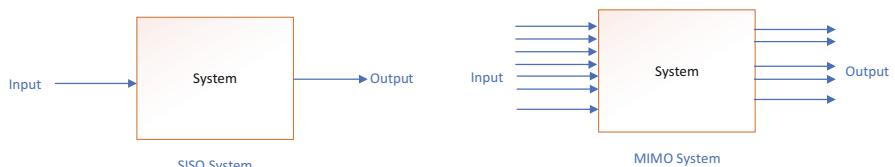


Fig. 2.6 SISO and MIMO model analysis

$$x_2[n+1] = y[n+2] = x_3[n]$$

$$x_1[n+1] = y[n+1] = x_2[n]$$

Thus the state equations are:

$$x_1[n+1] = x_2[n]$$

$$x_2[n+1] = x_3[n]$$

$$x_3[n+1] = -2x_3[n] - 5x_2[n] - x_1[n] = u[n]$$

The matrix form of the state equation is:

$$\begin{bmatrix} x_1[n+1] \\ x_2[n+1] \\ x_3[n+1] \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -5 & -2 \end{bmatrix} \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u[n] \quad (2.18)$$

The general form of the solution is

$$x[n] = A^n x[0] + \sum_{i=0}^{n-1} A^{n-1-i} b[i] u[i] \quad (2.19)$$

The discrete-time state space equations are written as follows. In discrete time state space equation, n indicates the present sample and $n + 1$ is the next sample.

$$x[n+1] = Ax[n] + bu[n]$$

The continuous-time state space equation of discrete-time state space equation is provided in 2.20:

$$y[n] = Cx[n] + du[n] \quad (2.20)$$

$$\begin{aligned} x[n+1] &= Ax[n] + bu[n], \text{ or} \\ \dot{x}[n] &= Ax(t) + bu(t) \end{aligned} \quad (2.21)$$

The system in Eq. 2.17 represented by state space representation is shown in Fig. 2.7.

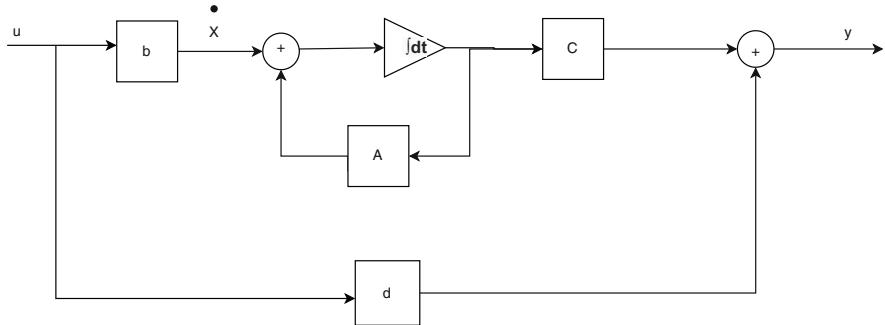


Fig. 2.7 The state space representation of a system

2.10 Common Measurements

Correlation, covariance, coherence, spectrum signal energy and power spectral density are basic and common signal measurements. The correlation and convolution are shift invariant and linear operations. These are introduced below.

2.10.1 Convolution

Convolution is a mathematical operation which combines two signals to generate the third one. In Eq. 2.22, observation s is mixed with desired input x and noise v . Here the operation is one signal x is flipped over the other signal h as shown in Eq. 2.23. The “ $*$ ” denotes the convolution symbol.

$$s[n] = x[n] * h[n] + v[n] \quad (2.22)$$

$$x[n] * h[n] = \sum_{i=-N}^N x[i]h[n-i] \quad (2.23)$$

Thus the convolution is a type of *FIR* filter. This definition may be more applicable for the deterministic case. The convolution also describes the statistical distribution of a system that has evolved from combining two isolated and distinct sub-systems characterized by specific statistical distributions. When the systems are combined linearly, the statistics of the observation is a normal distribution. This is due to the *central limit theorem* and the stochastic behavior of the system h .

The noise v , in Eq. 2.22, is unknown and the system h is approximated in terms of dirac delta function in a noise free environment ($v[n] = 0$, simplifying assumption). Thus,

$$s[n] = h[n] * \delta[n] = h[n] \quad (2.24)$$

In the previous equation, h is often referred to as the impulse response because it describes the response of a system to an impulse. The fundamental model has equivalence in frequency space written in Eq. 2.25. Here, S , X , H are the spectra of s , x , and h in the frequency ω . In the equation, h characterizes how the frequency distribution of the input is transferred to the output, and for this reason it is called as transfer function.

$$S(\omega) = X(\omega)H(\omega) \quad (2.25)$$

The definitions in Eqs. 2.23, 2.24, and 2.25 incline to the ideal noise free situation. But in practical situations, a process is mixed with a variety of noises. The noise is modeled applying statistical models that best fit to the noise, to the process, and to the scenario. This involves information extraction of the noise. The Bayesian estimation methods are one common approach to extract the information from noise. This is discussed in later chapters.

2.10.2 Correlation

Correlation between two signals generally tells us about the similarity between the two signals. This is also used to find a pattern of the same signal or different signals. Correlation is a preliminary approach that is used for random signal analysis. In Eq. 2.26, signal x is applied and shifted on y and it has $2N + 1$ elements from $-N$ to N . The symbol used in Eq. 2.26, \square is used here for correlation. This symbol is not really standard. The cross correlation is the correlation between two different signals. The cross correlation can be used to measure the template matching or when searching for the patterns.

$$C_{xy}(\tau) = x \square y(\tau) = \sum_{i=-N}^N x(i)y(i + \tau) \quad (2.26)$$

The auto-correlation is the correlation of the signal with itself. It can be used to measure repetitive patterns within a signal. It is applied, for example, to find the pattern in the musical beats. In Eq. 2.27, the autocorrelation of the x at some time lag τ can be depicted. Here the signal is being compared with the signal at certain time lag τ . One common application is a detection of the periodicity or periodic pattern of the signal. The auto-correlation of a periodic signal is periodic with the same period.

$$C_{xx}(\tau) = x \square x(\tau) = \sum_{i=-N}^N x(i)x(i + \tau) \quad (2.27)$$

At lag 0, the signal generally shows maximum auto-correlation. For periodic signal, auto-correlation function displays maximums and minimums according to its periodicity. Thus, detecting the periodicity is a common application of the auto-correlation. In addition the following applications can be found: Echo Cancellation, Aligning Two Simple Signals, Aligning Signals with Different Start Times, Finding Periodicity Using Auto-correlation, etc.

One common use of auto-correlation is short time Fourier Transform (STFT). In this case, N length of the signal s has maximal auto-correlation at $s(0)s(0)$ and the other terms are $s(1)s(0) + s(2)s(1) + s(3)s(2) + \dots + s(N-1)s(N-2)$. At zero lag, the signal s has $N-1$ terms to be auto-correlated, and at lag m , the signal s has $N-m$ terms to be correlated. The $s(0)s(0)$ term is the power of the signal s . This is shown in Eq. 2.28.

$$C_{ss}(m) = s \square s(m) = \frac{1}{N-m} \sum_{n=1}^{N-m} s(n)s(n+m) \quad (2.28)$$

The Wiener-Khinchin theorem is derived from cross correlation. The theorem says the Fourier Transform of the cross-correlation of a stationary signal with finite energy is the cross-energy spectral density. In Eq. 2.29, Fourier Transform denoted by FT of the auto-correlation of the signal F and G at frequency ω denoted by C_{fg} at certain time lag τ is the cross energy power spectral density denoted by P_{fg} at frequency ω of signals at the same frequencies.

$$FT [C_{fg}(\tau)] = F(\omega)G(\omega) = P_{fg}(\omega) \quad (2.29)$$

2.10.3 Auto Covariance

Signals are correlated with the preceding and following samples. Such structures are in statistics measured by co-variance and correlation, defined for zero-mean variables x and y . The auto-co-variance of the signal is commonly defined as:

$$r(k) = \sum_{t=-\infty}^{+\infty} x(t)x(t - k) \quad (2.30)$$

The auto-correlation is simply the covariance of two random variables such as $x(i)$ and $y(i + \tau)$, and it is autocovariance when the two random variables are of the same measured signal $x(i)$ and $x(i + \tau)$, or the correlation of the same random process x .

If the auto-correlation is C in Eq. 2.31 and auto-covariance is denoted by R , then

$$C_x x(i, j) = E[x(i)x^*(j)] \quad (2.31)$$

and the auto-covariance is given by Eq. 2.32. Some definitions of this use normalized versions of the signal where the mean, μ_x , is subtracted first.

$$R_{xx}(i, j) = E[\{x(i) - \mu_x(i)\}\{x^*(j) - \mu_x^*(j)\}] \quad (2.32)$$

The auto-correlation and auto-covariance are related by Eq. 2.33.

$$R_{xx}(i, j) = R_{xx}(i, j) - \mu_x(i)\mu_x(j) \quad (2.33)$$

In signal processing, the covariance is typically used as a coefficient between signals/data sets for similarity measurement; while the auto-correlation is used to characterize the correlation distance, namely, how quickly a signal evolves to another signal.

2.10.4 Coherence

The coherence is the measure of similarity between the signals in the frequency domain. It finds the common frequencies between the signals. Coherence is a normalized cross-spectral density function that is used to measure the signal to noise ratio (SNR) between sensor arrays.

The $P_{xx}(\omega)$ and $P_{yy}(\omega)$ are the power spectrum of the signals x and y in Eq. 2.34, while ω is indicating the frequency domain of the signals. The $P_{xy}(\omega)$ is cross power spectrum of the both signals x and y .

$$C_{xy}(\omega) \equiv \frac{P_{xy}(\omega)}{\sqrt{P_{xx}(\omega)P_{yy}(\omega)}} \quad (2.34)$$

Measurement of SNR, correlation with respect to frequency, array gain, system identification, and determination of time delays use the coherence function. The array gain is the coherence of the signal and the noise between sensors of arrays. The coherence of the signal between sensors improves with decreasing distance between the sensors, frequency of the received signal, total bandwidth, and integration time. Example including applications and more theoretical foundation can be found in [Pouliarikas09].

Question 2.7 If the input is $(1, 1, 1, 1, 1)$ and the impulse response of a system is $(1, 0.5, 0.2)$, what is the output?

Solution This can be done by a multiplication of the input and system as polynomials. If we arrange the input as $X(z)$, the impulse response as $H(z)$, and the output is

$Y(z)$ such that $X(z)$ is equal to $X(z) = 1 + 1z^{-1} + 1z^{-2} + 1z^{-3} + 1z^{-4}$, and $H(z)$ is given by:

$$H(z) = 1 + 0.5z^{-1} + 0.2z^{-2} \quad (2.35)$$

The output is obtained as:

$$Y(z) = X(z) \cdot H(z) \quad (2.36)$$

where “.” implies multiplication. Then $Y(z)$ is:

$$Y(z) = (1 + 0.5z^{-1} + 0.2z^{-2})(1 + 1z^{-1} + 1z^{-2} + 1z^{-3} + 1z^{-4}) \quad (2.37)$$

$$Y(z) = (1 + 1.5z^{-1} + 1.7z^{-2} + 1.7z^{-3} + 0.5z^{-4} + 0.7z^{-5} + 0.2z^{-6}) \quad (2.38)$$

Then the output signal is $(1, 1.5, 1.7, 1, 7, 1.7, 0.7, 0.2)$ This is the result of convolution operation of the two signals in the frequency, namely $X(z)$ and $H(z)$ resulting in $Y(z)$.

The graphical representation of results is depicted in the Fig. 2.8.

2.10.5 Power Spectral Density (PSD)

The representation of the digital signal in terms of its frequency components in a frequency domain is called the signal spectrum. The Fourier transform of the auto-correlation is the power spectral density. The Fourier spectrum of a signal $x[n]$ where $n \in Z$ and Z denotes the set of any integer numbers in $X(f)$.

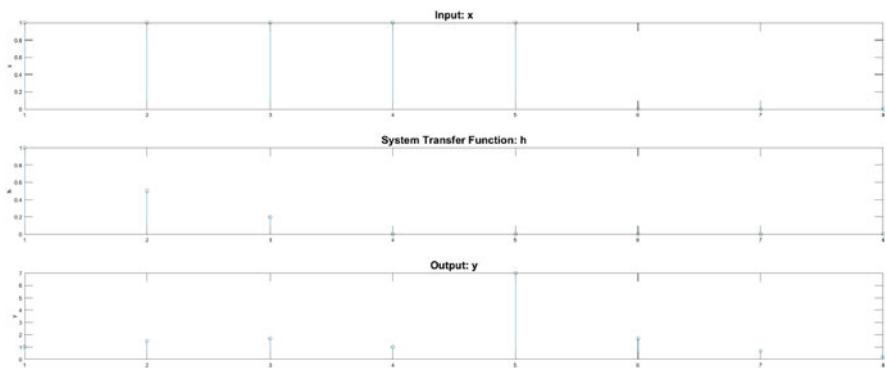


Fig. 2.8 Input, system and output signals

$$X(f) = |X(f)|e^{(j\theta(f))} \quad (2.39)$$

In Eq. 2.39, $|X(f)|$ is the amplitude spectrum and $\theta(f)$ is the phase spectrum. The power spectrum of signal $x[n]$ is $|X(f)|^2$.

The amplitude and power spectrum of the signal is often measured in decibel dB scale. The dB scale is a logarithmic scale in which the order of the magnitude changes in powers of 10 dB and the order of the magnitude in amplitude changes by 20 dB . The amplitude and power in decibels are expressed as the relative to a reference signal.

An average power P of deterministic finite signal $x_T(t) \forall -T < t < +T$ is in Eq. 2.40. Now $x(t)$ is considered power signal only if P is finite, that is $P < \infty$.

$$P = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} |x_T(t)|^2 dt \quad (2.40)$$

The power of the signal $x_T(t)$ is provided in Eq. 2.42 where $X_T(f)$ is the Fourier transform (FT) of the signal $x_T(t)$.

A square of the absolute valued signal gives the energy of a signal. In Eq. 2.41, E is the energy of the signal.

A signal $y(t)$ is an energy signal if and only if $0 < E < \infty$. An energy signal has non-zero, finite energy and zero power:

$$E = \sum_{-\infty}^{+\infty} |y(t)|^2 \quad (2.41)$$

The signal energy is a common speech feature component. Similarly, a power signal has non-zero, finite power and finite energy. A signal $y(t)$ is a power signal if and only if $0 < P < \infty$:

$$\begin{aligned} P &= \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-\infty}^{+\infty} |x_T(t)|^2 dt = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-\infty}^{+\infty} |X_T(f)|^2 df = \\ &\quad \int_{-\infty}^{+\infty} \left(\lim_{T \rightarrow \infty} \frac{1}{2T} |X_T(f)|^2 df \right) \end{aligned} \quad (2.42)$$

The power spectral density of the signal $x_T(t)$ is given in the next Eq. 2.43:

$$P = \int_{-\infty}^{+\infty} |S_x(f)| df \text{ where } S_x(f) = \lim_{T \rightarrow \infty} \frac{1}{2T} |X_T(f)|^2 \quad (2.43)$$

For infinite energy signal it may not have power spectral density. For the random signal, PSD is calculated by computing the expectation of the signal. If $X(t)$ is a wide-sense stationary random process (WSS), then the expectation of the signal is given in Eq. 2.44:

$$E[x(t)] = \mu_x \quad (2.44)$$

The power of the random process is normally calculated by Parseval's or Rayleigh theorem.

$$P_{xi} = \int_{-\infty}^{+\infty} |X_{Ti}(f)|^2 df \quad (2.45)$$

For a WSS random process, the FFT of the auto-correlation function and the power spectral density are equivalent, and this is called Wiener-Khinchine theorem. This means the PSD of the signal can be computed if the auto-correlation is known. The power spectral density of a random process $x(t)$ is computed in Eq. 2.46 where $S_{xx}(j\omega)$ is the continuous-time Fourier transform (CTFT) of the auto-correlation function $R_{xx}(\tau)$.

$$E[x^2(t)] = R_{xx}(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{xx}(j\omega) d\omega \quad (2.46)$$

The variation of the stochastic event is uncertain and therefore not known. One common approach to characterize the process is probabilistic measurements. Here the probability of the uncertain events is evaluated by considering the ensemble of the process. They do not have finite energy and hence the DTFT cannot characterize the signal. The process has finite average power, and hence the power spectral density (PSD) is one common primary approach to characterize the signal.

$$\varphi(\omega) = x(0) + 2 \sum_{k=1}^{+\infty} \operatorname{Re} \left[x(k) e^{(-i\omega)} \right] \quad (2.47)$$

In Eq. 2.47, the PSD $\varphi(\omega)$ measures the power at frequency ω of the spectrum of the signal's auto correlation function.

Question 2.8 A random process $Y(t) = X(t) + X(t - T)$ where $X(t)$ is a wide-sense stationary random process with auto-correlation function $R_x(\tau)$ and power spectral density $S_x(f)$. Find $R_y(\tau)$ in terms of $R_x(\tau)$.

Answer:

$$\begin{aligned} R_y(\tau) &= E[Y(t)Y(t - \tau)] \\ &= E[\left(X(t) + X(t - \tau) \right) \left(X(t - \tau) + X(t - \tau - T) \right)] \\ &= E[X(t)X(t - \tau)] + E[X(t)X(t - \tau - T)] \\ &\quad + E[X(t - T)X(t - \tau)] + E[X(t - T)X(t - \tau - T)] \end{aligned}$$

$$R_y(\tau) = 2R_x(\tau) + R_x(\tau + T) + R_x(\tau - T)$$

$$\begin{aligned} S_y(\tau) &= F\{R_y(\tau)\} \\ &= F\{2R_x(\tau) + R_x(\tau + T) + R_x(\tau - T)\} \\ &= 2S_x(f) + S_x(f)e^{(j2\pi fT)} + S_x(f)e^{(-j2\pi fT)} \\ S_y(\tau) &= S_x(f)[2 + 2\cos(\pi fT)] = 4S_x(f)\cos^2(\pi fT) = F \end{aligned}$$

2.10.6 Estimation and Detection

The estimation and detection is mainly based on statistical measurements and decision making. Some common approaches are hypothesis testing for example binary hypothesis testing, M -ary tests, for the optimal or robust detection and classical estimation techniques such as best linear unbiased estimator (BLUE), maximum likelihood estimator (MLE), least squares estimation (LSE), expectation-maximization (EM), Bayesian estimation, maximum a posteriori (MAP) estimation, minimum mean square error (MMSE) estimation, linear minimum mean square error (LMMSE) estimation, some iterative estimation techniques such as least mean squares (LMS) algorithms and recursive least square (RLS) algorithms. These are some estimation techniques that are commonly used for statistical signal or data estimation. Kalman filtering, Kalman-Bucy filtering, Wiener filtering, Particle filtering are also some statistical estimators are commonly used in adaptive filtering.

A simple example of signal detection and estimation is in warship tracking. On processing the message of the arrival of a warship or submarine, the location of the warship or the submarine is an example of signal estimation. Two common estimation types are parameter estimation and state estimation. In parameter estimation, the best approximates of the parameters of the model are investigated given a set of observations leading to, for example, a probabilistic model. For instance, the speech is estimated by LPC parameters estimation. In state estimation, the best estimate of the state is investigated based on the prior information in the form of a model.

For example one has the Bayesian estimation. In this procedure, the parameter estimation is used in order to get the approximated parameters in the prior model. Example of such state estimation is the Kalman filtering. The detection is estimating the best approximation among the estimated results. For example one considers the bit detection at the receiver. In some sense, the estimation is a continuous hypothesis where detection is the discrete hypothesis. Different estimation types are discussed in the adaptive filtering chapter and the applications of the different estimations is given in Part III.

In estimation, two approaches are common such as parameter estimation and non-parameter estimation. The parameter estimation is Bayesian estimation and

maximum likelihood (MLA) where the techniques in non-parametric estimation are Kernel Density estimation and nearest Neighbor rule. This will be discussed later on in the connection to similarities.

The detection and estimation theory is widely used in many areas for information extraction or to find a proper interference in the process to be controlled from the desired one. Examples of such areas are radar, communication, seismology, sonar, speech and image processing, medical and biomedical signal processing.

The detection problem can also apply hypothesis testing such as Bayesian hypothesis testing, minimax hypothesis testing, Neyman Pearson hypothesis testing, composite hypothesis testing. Each of these hypothesis testing is applied on certain individual application types. Example of such hypothesis testing in detection theory is discussed in Chap. 6. While estimating, we are interested in estimating models θ from some x as in following equation:

$$y_t \approx f(\theta, x_t) \quad (2.48)$$

2.10.7 Central Limit Theorem

The central limit theorem states that the sum of an independent, identically distributed (IID) random process with zero mean finite variance converges to the distribution of a Gaussian random variable. For instance, an IID random process $\{X_n\}$ has the sequence $\{x_1, x_2, x_3, \dots, x_n, \dots, x_L\}$ of size L . The expected value of this process is μ which is zero or zero mean and the variance is σ^2 . Thus the process $\{X_n\}$ can be described as $\{\{X_n\}\} \{X_n\} = N(\mu = 0, \sigma^2)$ as shown in Eq. 2.49.

$$\{X_n\} = \frac{1}{\sqrt{N}} \sum_{n=1}^L x_n \rightarrow N(\mu = 0, \sigma^2) \quad (2.49)$$

According to the central limit theorem, if the sample is sufficiently large, then the expectation of the random process has a sampling normal distribution, regardless of the shape of the process. The estimation is better for the larger sample sized process (Fig. 2.9).

Question 2.9 The average GPA at a particular school is $\mu = 2.89$ with a standard deviation $\sigma = 0.63$. A random sample of 25 students is collected. Find the probability that the average GPA for this sample is greater than 3.0.

Answer

The standard error is $\epsilon = \frac{\sigma}{\sqrt{n}} = \frac{0.63}{\sqrt{25}} = 0.126$

The z-score is $z = (3 - 2.89)/0.126 = 0.11/0.126 = 0.87$. The z-score in the normal curve table yields probability of 0.8078. Thus the probability is $1 - 0.8078 = 0.1922$.

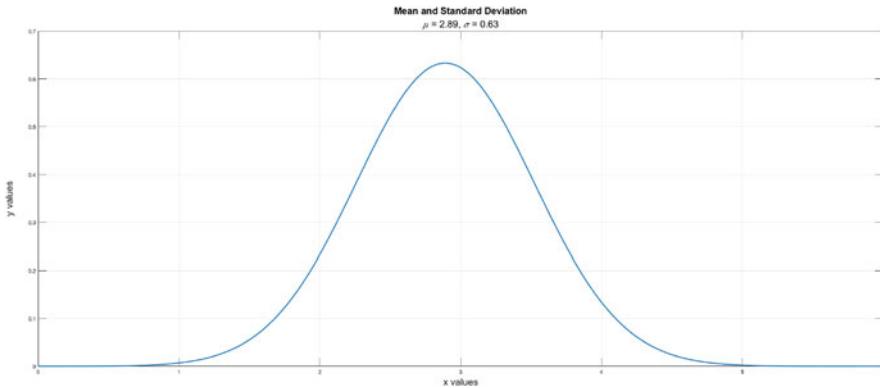


Fig. 2.9 Standard error

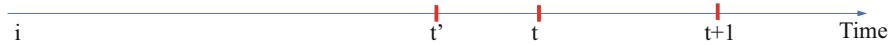


Fig. 2.10 Time range

2.10.8 Signal Information Processing Types

Signal processing deals with the information processing and manipulation of the signal information. Data fusion combines information from multiple receivers and transfer the information to a common control unit to be combined and processed and fused to a decision that can be relayed to each of the receivers. The receivers for the data fusion can be located at different ranges in different media using different mechanisms. Three of the most common types are:

- Filtering: Here the range of signal used in filtering is known up to a certain time t at which state parameters are estimated.
- Prediction: Here the range of the signal is known up to some previous time before t , and used to predict the state parameters at time t .
- Smoothing: Here the range of signal is known until some time t , and is used to infer information about state parameters at an intermediary time prior to t .

In the Fig. 2.10 the smoothing range of the signal is i to $t + l$ and it is used to infer the state parameters at times t' .

2.10.9 Machine Learning

Machine learning is an area where a machine or the computer mimics the real world by training and learning. For example, speech recognition is an area where the study of signal processing and machine learning are applied. In speech recognition, in

general, a machine or a computer is trained to mimic the interaction of a human being. It is used, for example, to perform some specific tasks where the speech is the input and the output can be speech, or text, or image, or some other forms of communications. Machine Learning is a task to obtain some unknown property of certain processes in some set. These processes contains the original signals and one wants to know the information on a higher level of abstraction. In real world, there are frequently unexpected, random, unknown problems where the information such as data are unstructured and contain hidden information. Often the real world problems are not closely defined and the problems are approached as an open world problems. Solutions of these problems are dealt with by defining them as dynamic processes, work flows, actions, techniques, states, parameters, features, etc. One of the difficulties of such problems are unknown, unpredictable structures, hidden information. Further solutions of these problems give rise to learning. Exploring such hidden information and revealing it is also studied in data mining and in this sense machine learning may be considered as a sub area of data mining. An overview of how to handle such problems is shown in Fig. 2.10.

These properties can be relevant, for instance, in decision making, diagnostics, information retrieval, image analysis and image retrieval.

In the context of the book the reader may be interested in properties of signal processes. These properties are not directly visible and hidden in the many signals of a signal process. The machine learning is a (sometimes elementary) step in recovering properties of interest and machine learning tools are used for this purpose.

We apply machine learning to learn the data, to extract patterns, analyze them, finally test them to make decisions. We apply machine learning tools to find good models for the best predictive and decision making capabilities with least errors.

Visual analytics (VA) applies machine learning tools to study what the data is, where we find it, and how can we explore it. In VA we learn the data by exploring, predicting, and inferring. VA can be descriptive, and predictive where one uses data aggregation and data mining to provide an insight into the past and find an answer, i.e., “What has happened, when it is?” VA can be accomplished in various ways. Common step studied in the text include exploratory data analysis, feature generation, extraction, preprocessing, various model validation techniques, data leakages, metrics optimization, model assembling and hyper-parameter tuning. Real world data analysis requires understanding the problem, problem formalization, data collection, data processing, and modeling. Analysis has the relationship of the attributes or the features, grouping, identifying the patterns, possible methods to be applied for the transformation.

Machine learning, knowledge discovery, natural language processing, and information retrieval are used to design computational models for automated text analysis and mining. Major areas in text mining are:

- Keyword extraction
- Classification and clustering

2.10.9.1 Anomaly and Trend

One of the major focuses in machine learning is the discovery of the patterns of the data. These can be in different shape, size, and types. For example, data that expresses an image, data that expresses the sound, data that expresses the stock prices, data that expresses the seismic evolution, data that is supposed to be presented on the web for their visualization, characteristics, or for any business or in building an e-commerce website. Similarly all these different data types can be with different dimensions. For example the image data can be two dimensional, and again this can be multidimensional. The data can be binary, digital or discrete or analog. Regardless of the types, varieties, and shapes, data has to be characterized, organized and patterned for machine learning and recognition.

Theoretical aspects are discussed in Part **II**, and applications relating to signal processing and machine learning are provided in Part **III**.

In classification, the inputs are assigned to one of two or more classes. A decision rule divides the input space into decision regions separated by decision boundaries. There are many different types of classifiers for classification and recognition of data. Some examples are *K-nearest neighbor*, *support vector machine (SVM)*, *decision trees*, *neural networks (NN)*, *naïve Bayes*, *Bayesian network*, *logistic regression* etc. The tasks of the classification can be generative or discriminative or a combination of the two. In generative classification, the data are modeled and labelled. Statistical information such as conditional probability and prior information are used in the classification. An example of such generative classification is *Bayes classification*. In discriminative classification, the data is labelled and the data is learned from these labels. Examples of such classification algorithms are *SVM*, *logistic regression*, and *decision trees*.

Multivariate data analysis and similarity measurement is one of the common tasks in machine learning. The data can be input for the machine analysis as *input vector*, *pattern vector*, *feature vector*, *sample*, *training set*, *reference set* or *test set*. They might be called as features, components, or attributes. These inputs can be *ordered*, *unordered*, *semi ordered* in a *discrete*, *real*, or *categorical* manner.

Regardless of the types, inputs, pattern, the learning can be of three types, such as *unsupervised learning* where the pattern of the data is investigated and the input is only unlabelled data while a goal of this training is discovery of that pattern. The second is *supervised learning*, it learns the data using labels or categories, and the *semi-supervised* uses or *reinforced learning* is in between supervised and unsupervised learning that uses both.

Next in Fig. 2.11, we see the structure of an approach solving a typical machine learning problem, but details may vary according to problems and applications.

2.10.9.2 Background Information

The signal processing can be viewed as a refining the information in the signal as required. A stochastic process loosely means that the measured signal is different,

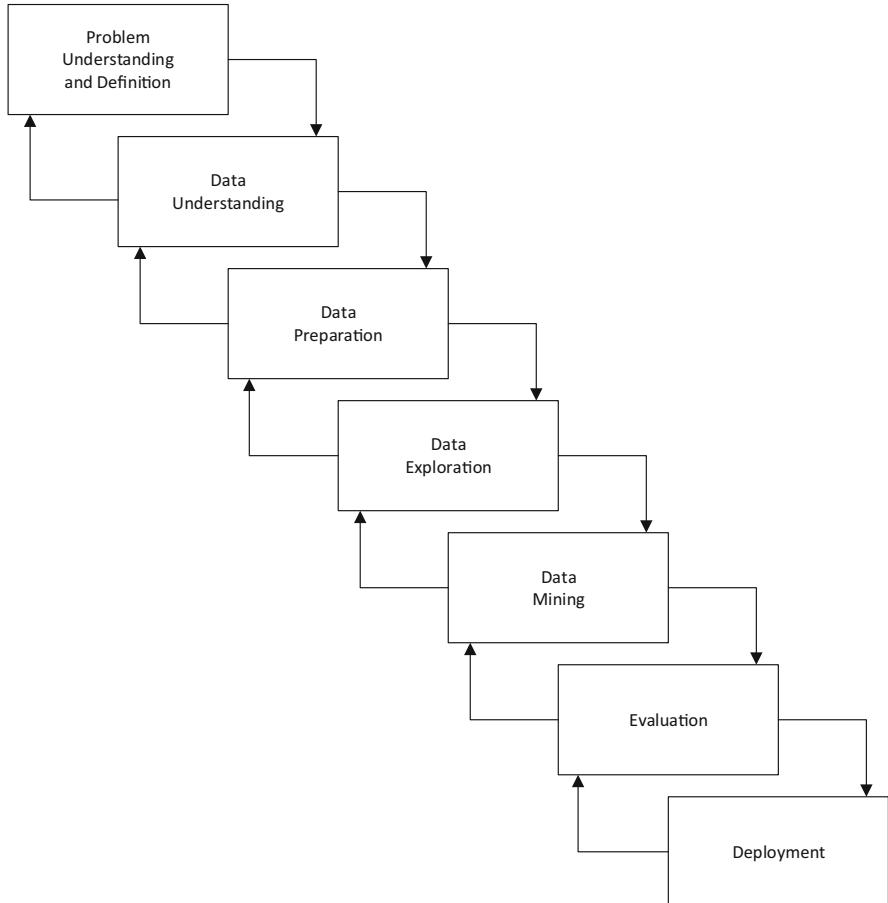


Fig. 2.11 Exploring problems in machine learning

consequently measurements would be different as experiments are performed. A random process can be a synonym of this. Signal processing and communications are inter-related with signals and systems. Time, frequency, adaptive equalization, blind equalization, filtering, adaptive beam forming, signal intelligence, spectral sensing, estimation, detection, identification, information and communication theory are topics in signal processing. The area is vast and almost all applications are rooted on this area. A stationary signal indicates same statistical properties of the signal and it remains constant or the same over time. An observed record of a random or stochastic process is merely one of a whole collection of records that could have been observed. The collection of all possible records is called the ensemble and an individual record is a realization of the process. One experiment gives a single realization. Various realizations are $X(n, \omega_1)$, $X(n, \omega_2)$, ..., but the

fact that generally only a single realization is available gives the possibility of dropping the argument ω .

Signal processing is a bridge between the information and the applied mathematics. The signal processing has enormous applications behind the digital society. The mobile phone, telephone, television are example areas of everyday life where signals are in the center of interest. Many applications are presented in Part III. For instance, in Chap. 21 a focus is on applications of speech signal.

The machine learning is an area in its own and will be discussed extensively in this book. Related lecture notes can be found in (Richter 2008). We are not so much interested in theoretical backgrounds but rather in aspects that are connected with applications. A list of recommended text is provided in reference.

Readers are encouraged to consult references used in this chapter for example, Rao and Yip (1990), Keller (2004), Addison (2002), Strang (1993), Byrne (2013), Smith (2011), Meyer (1992), Wesfried (1993), Nason (2008), Diniz (2001), Selesnick (2007), Brown (2018), Mitchell (1997), Langley (2011), Miller (1960), Amari (1996), Poularikas (2009), Barten (1982), Intriligator (1987), Steven (2012), Syed (2015), and Piet (2006).

2.10.10 Exercises

Question 2.10 Derive signal expression for the exponential signal $U(t) = e^{-at}$ for $t \geq 0.0$ in frequency domain?

Question 2.11 Assume channel impulse response $(1, 0.5, 0.2)$. Show each received signal for:

1. $X_1 = [1, 0, 0, 0, 0, 0, 0]$
2. $X_2 = [1, 1, 1, 1, 1, 0, 0]$
3. $X_3 = \cos(2 * \pi * n / 32); N = 1 : 100$

Question 2.12 Smoothing Data by N point convolution: Save the data file *data_emg.txt* and a *speech.txt*. The data are collected from measuring electrical signal produced by speech spoken by a speaker. Plot the signals. The impulse response for an LTI system is: $h = \text{ones}(1, 11) / 11$;

1. Compute the convolution of h and x , where $x = \text{data_emg.txt}$
2. Compute the convolution of h and y , where $y = \text{speech.txt}$

Explain:

- (a) How does the convolution change x and y ? Compare the change with the input and output, that is, x and y ?
- (b) How the length of the output of the convolution of h and x as well as h and y related?

- (c) Repeat the problem with a different impulse response (e.g., different system), such as $h_1 = \text{ones}(1, 31)/31$ and explain as well as comment on the observation.
- (d) Repeat the problem with a different impulse response (e.g., different system), such as $h_2 = \text{ones}(1, 67)/67$ and explain as well as comment on the observation.

References

- [Rao1990] K. R. Rao and P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications, Academic Press, Boston MA, 1990
- [Keller2004] Keller, Wolfgang. Mathematik: Wavelets in Geodesy and Geodynamics: Walter de Gruyter, 2004
- [Addison2002] Addison, Paul S. The Illustrated Wavelet Transform Handbook, IOP Publishing Ltd, UK, 2002
- [Strang1993] Strang, Gilbert. Wavelet Transforms Versus Fourier Transforms, American Mathematical Society, Vol 28, Number 2, April, 1993
- [Byrne2013] Charles L. Byrne, Signal Processing : A Mathematical Approach CRC Press, Second Edition, Nov 12, 2014.
- [Smith2011] Smith, Steven W. The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, 2011
- [Meyer1992] Meyer, Y., Wavelets and Operators. Cambridge: Cambridge University Press. ISBN 0-521-42000-8., 1992
- [Wesfried1993] Eva Wesfried, E. and Mladen Victor Wickerhauser, M. V., Adapted Local Trigonometric Transforms and Speech Processing, IEEE Transactions on Signal Processing 41(12), October, 1993
- [Nason2008] Nason, G., Wavelet Methods In Statistics With R, Springer, August 2008
- [Diniz2001] Diniz. P S. R. Digital Signal Processing: System Analysis and Design, Poet Chester, NY, USA, Cambridge University Press, 2001
- [Selesnick2007] Selesnick, I., Wavelets, a Modern Tool for Signal Processing, Physics Today. 60(10):78-79, October 2007
- [Brown2018] Brown, Peter C and Roediger III, Henry L, CITIC Press, 2018
- [Mitchell1997] Mitchell, Tom M and others, Machine learning, McGraw-hill New York, 1997
- [Langley2011] Langley, Pat, The changing science of machine learning, Springer, 2011
- [Miller1960] Miller, George A and Galanter, Eugene and Pribram, Karl H; Plans and the structure of behavior, MillerPlans and the Structure of Behavior, 1960
- [Amari1996] Amari, Shun-ichi and Cichocki, A. Yang, H. H., and others; A new learning algorithm for blind signal separation (Chapter), Advances in neural information processing systems (BookTitle) pages(757–763), Morgan Kaufmann Publishers, 1996
- [Pouliarikas09] Koch, W. (Chapter Author), A. D. Pouliarikas, A. D. (Editor), ADVANCED SIGNAL PROCESSING Theory and Implementation for Sonar, Radar, and Non-Invasive Medical Diagnostic Systems, CRC Press, 2009
- [Barten1982] Barten, A. P. and Böhm, V., Consumer theory: Handbook of mathematical economics, pages (381–429), vol(2), Elsevier, 1982
- [Intriligator1987] Intriligator, M. D. Arrow, Kenneth J, Handbook of mathematical economics, North Holland, 1987

- [Steven2012] Steven K. T., Signals and Systems with MATLAB Computing and Simulink Modeling, 5th edition, Orchard Publication, 2012
- [Syed2015] Syed, A. H., Adaptation and Learning, Lecture Notes, 2015
- [Richter2008] Richter, M. M., Lecture notes in Machine Learning and Business Processes, University of Calgary, Canada, 2008
- [Piet2006] Broersen, P. M. T., Automatic Autocorrelation and Spectral Analysis, Springer-Verlag, 2006

Chapter 3

Fundamentals of Signal Transformations



Overview

In this chapter, we provide an overview of mathematical transformations, representations, and descriptions of the signals. The signal transformation varies with the time, and space, and depends on its continuity. For analysis purposes, the signal is transformed into a set of functional blocks. The function is a mapping or a relation from one form to another form. Outline of different transformations and how they are applied to signals is given in order to process them. The signal transformations can be different based on applications. Some linear, nonlinear or a combination of basic functions transforms the signals. These basis functions can result from Fourier transform, or Wavelet transform or the cosine transform. An important transformation is feature extraction. The Fourier transform analyzes constituent components of the signal. But most real world signals are not periodic, rather they are non-stationary. The **Short Time Fourier Transform** (*sft*, or *STFT*), **wavelet** transform, and **Local Cosine Transform** (*LCT*) are such transforms that analyzes signals locally with respect to time and frequency, and also adaptively. Specifically the wavelets transform and *LCT* analyzes the signals adaptively and locally with respect to time and frequency.

The Laplace transform is defined below:

$$x(t) \leftrightarrow X(s)$$

$$X(s) = \int_{-\infty}^{+\infty} x(t)e^{-st} dt$$

In continual-time Fourier transform (*CTFT*), the following definitions is applied:

$$x(t) \leftrightarrow X(j\omega)$$

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt$$

In discrete case, the z -transform is applied:

$$x[n] \leftrightarrow X(z)$$

$$X(z) = \sum_{n=-\infty}^{+\infty} x[n]z^{-n}$$

In discrete-time Fourier transform (*DTFT*), the signal is discrete in time, however, its transform is continuous, hence the emphasis in the name (e.g., discrete-time),

$$x[n] \leftrightarrow X(e^{j\Omega})$$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\Omega n}$$

In discrete Fourier transform (*DFT*), the discrete-time representation is further discretized in frequency through N points:

$$x[n] \leftrightarrow X(k)$$

$$X(k) = \sum_{n=0}^{N} x[n]e^{-j2\pi \frac{kn}{N}}$$

In short time Fourier transform representation is defined as:

$$x[f, s] = \int_{-\infty}^{+\infty} x(t)g(t-s)e^{-j2\pi ft} dt$$

where $g(t)$ is the windowing function that is used to limit the range of the observed data. Similarly, in the wavelet transform presented below, it further generalizes the Fourier transform as given below:

$$x[u, s] = \int_{-\infty}^{+\infty} x(t)\varphi_{u,s}(t)dt \quad \forall \quad u \& s$$

Note that u indicates the translation and s indicates dilation. In local trigonometric transformation (*LTT*) transform the following is being defined:

$$x(t) \leftrightarrow \sum_{j \in Z} \sum_{k \in N} c_k^j \phi_k^j(t)$$

where $c_k^j = \langle x(t), \phi_k^j(t) \rangle$, and that is further decomposed as follows:

$$c_k^j = \sqrt{\frac{2}{|I_j|}} \int x(t) b_j(t) \cos \frac{\pi}{|I_j|} (k + \frac{1}{2})(t - a_j) dt$$

where $b_j(t)$ is characteristic function of the finite interval, I_j .

The Laplace transform finds the pole and zero representation of the continuous-time, $x(t)$ in the s -plane, the z -transform finds the pole and zero representation of the discrete-time system $x[n]$ in the z -plane, the *CTFT* is the spectral analysis using complex exponential form of the continuous-time signal. In *CTFT*, $s = j\omega$. The *DTFT* stands for discrete time Fourier transform and the *DTFT* is continuous in the frequency domain such that $z = e^{j\Omega}$. The *DFT* is the discrete Fourier domain representation of signal. In *STFT*, the window $g(t)$ as sliding along with signal $x(t)$ and each shift is $g(t - s)$. The Fourier transform of the product i.e. $x(t)g(t - s)$ is computed. The wavelet analysis is one of the most effective used tools that can capture signal's significant points such as trends, hidden periodicity finding, breakpoints, or discontinuities. The *STFT* also does local analysis of signals, where the wavelet does local and multiresolution analysis of signals. Fourier analysis uses sines and cosine function, wavelet uses basis functions. In the Fourier transform, changes in frequency cause changes in the overall in the time domain. Wavelets are local in both frequency and time using scale or dilations and translations.

3.1 Transformation Methods

Transformations occur frequently when dealing with signal processes. Below we give an overview of some commonly used transformations:

- Laplace transform
- Fourier transform
- Z-transform
- Discrete Cosine Transform (DCT),
- Wavelet transform

The details of the transformations discussed in this chapter can be found in references provided at the end of the chapter. Here we just give an essence of various types of transformations and we outline here how these transformations are applied to signal analysis.

A mathematical description of the input and output of a system can be analyzed in the time domain and in the frequency domain. The time domain analysis is based on differential/difference equations. This gives the system output as a weighted combination of the differentials/differences of the system input and output signals. The frequency domain analysis describes a system in terms of its individual frequency components of the input signal to their response. The common frequency analysis methods are *Laplace* transform, *Fourier* transform and the *Z*-transform. These transforms are complex exponential as their basis functions. The wavelet transform and conjugate wavelet transform such as the local trigonometric transformation (*LTT*) are adaptive signal transformations. The frequency analysis also describes the behavior and stability of a system. The integration and differentiation of the frequency variable in the frequency domain is another way to analyze the system in the frequency domain. In this chapter we also mention the linear time invariant system (*LTI*), and its application for designing a digital filter. Poles and zeros are the system or the filter coefficients for the system $H(z)$ or $h(n)$ or $H(\omega)$ where $H(z)$, $h(n)$ and $H(\omega)$ represent the z-transform, time-domain and frequency representations of the system. Then again a system is characterized by magnitude response, frequency response, gain, group delay and phase delay which are denoted by $|H(\omega)|$, $H(\omega)$, G , τ_g , τ_p .

3.1.1 Laplace Transform

The Laplace transform is used to analyze the frequency information of the continuous time signal. The Laplace transform is normally a one-sided Fourier transform with an exponential attenuation term. The Laplace transform is widely used in control system and system modeling. In Eq. 3.1 presented below, $f(t)$ and $F(s)$ are the time and s-domain representation of the signal. The Laplace transform analyzes the signal in terms of sine and cosine waves that have an exponentially changing amplitude. $L\{F(s)\}$ is the Laplace transform of the time domain.

$$L\{f(t)\} = F(s) = \int_{-\delta}^{+\delta} f(t)e^{-st} dt \quad (3.1)$$

The Laplace transform changes the signal specified in the time domain into a signal in the s-domain or s-plane. The time domain signal is continuous and extends in both positive and negative infinity and can be periodic or aperiodic. The time domain may be complex. The s-plane is a complex plane where the real numbers are depicted along the horizontal axis and imaginary number components along the vertical axis. The distance along the real axis is expressed by the variable σ and the imaginary axis is the frequency variable ω . The location of any point or a coordinate in the s-plane is denoted by σ and ω such as $s + j\omega$. For example, in Eq. 3.1, a time domain signal $f(t)$ is, in the s-domain, denoted by $F(s)$ expressed also as $F(\sigma, \omega)$. The s-plane is continuous and extends to infinity in all directions. Each co-ordinate in

the s-plane has real and imaginary components. A location of a point in the Laplace domain i.e. s-plane is denoted by a complex number that has its real and imaginary part (Cartesian representation). Alternatively each co-ordinate system is represented by magnitude and phase (polar representation).

The Laplace transform can be analyzed as either one-sided or two sided which are known as unilateral and bilateral. Equation 3.1 is two-sided while Eq. 3.2 is a one-sided Laplace transform (*LT*) of $f(t)$.

$$L\{f(t)\} = F(s) = \int_0^\delta f(t)e^{-st}dt \quad (3.2)$$

In Eq. 3.3, $L^{-1}\{F(s)\}$ is the inverse Laplace transform:

$$L^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j} \int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{st}ds \quad (3.3)$$

The Laplace transform is generally not used in signal processing applications and is not discussed here further.

3.1.2 Z-Transform

The Laplace transform, applied to continuous signals, is a differential equation, and z-transform is a differential equation that is applied to a discrete signal. The z transform expresses a discrete time signal into the z-domain. The z-transform yields a frequency domain description for discrete-time signals and forms the basis of the design of digital filters. The z-transform of the signal is written in Eq. 3.4.

The transform representation is:

$$H(z) = 1 - az^{-1} \quad \text{for } 0 \leq a \leq 1 \quad (3.4)$$

The z-transform of the signal $f[n]$ is written in Eq. 3.5. In the equation, $f[n]$ is the discrete time signal of the continuous time signal $f(t)$. Instead of the integral used by Laplace transforms in Eq. 3.1, z-transforms use a summation and the time variable t is replaced by sample number n, as shown in Eq. 3.5. In this equation, $F(\sigma, \omega)$ is still a continuous function and its parameters σ and ω have continuous values as the signal $f(t)$. With Laplace σ can take negative values i.e. $-\sigma$, it can take positive values $+\sigma$ and σ can be also 0. This controls how many samples n will be for a particular value of the signal $f(t)$.

$$F(\sigma, \omega) = \sum_{n=-\infty}^{\infty} f(n)e^{-\sigma n}e^{-j\omega n} \quad (3.5)$$

Now, one replaces $e^{\sigma n}$ by r^n which is shown in Eq. 3.6. Here r controls the decay of the waveform $f[n]$.

$$r^n = [e^{\ln(r)}]^n \quad \text{where } e \quad \sigma = \ln(r) \quad (3.6)$$

The Eq. 3.5 can be written in an exponential form shown in Eq. 3.7.

$$F(\sigma, \omega) = \sum_{n=-\infty}^{\infty} f(n)r^{-n}e^{-j\omega n} \quad (3.7)$$

Similar to the Eq. 3.1 written in the s -plane, Eq. 3.8 is shown now in z -plane where in Eq. 3.1, $s = \sigma + j\omega$ and Eq. 3.7, z -plane is used by setting $z = re^{j\omega}$. The z -transform of the $f(t)$ is shown in Eq. 3.8.

$$F(z) = \sum_{n=-\infty}^{\infty} f(n)z^{-n} \quad (3.8)$$

In many situations, the continuous system is described by differential equations and discrete systems are described by difference equations. The discrete systems can be analyzed by computing the transfer function and then computing the frequency response by pole zero diagram and frequency response. These are also used to analyze the recursion coefficients for filters such as the finite *impulse response (FIR)* filter and *infinite response (IIR)* filter.

The *roots* of the transfer function are the *poles* and *zeros*. The poles and zeros indicate the behavior of the systems. The poles generally represent the feedback part of the system while the zeros generally represent the feed forward part of the system. The steady and transient state of the system can be analyzed by poles and zeros.

In Eq. 3.9 $y(n)$ is the output and $x(n)$ is the input, a and b are the coefficients of the input and output. This equation is a time-domain representation of the signal.

$$y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k) \quad (3.9)$$

If we take the z -transform of the given expression, we obtain the Eq. 3.10. In this equation, $Y(z)$ and $X(z)$ are the z -transforms of the input and output. Suppose the system is a linear, time invariant, and causal system. We use the system equations to define the digital filter by first transforming this into a transfer function as shown in Eq. 3.10. Here we consider that the initial conditions of the system are fulfilled.¹

¹ The properties and theorems of z -transform are not discussed here. There are many texts covering this topic.

$$Y(z) = \sum_{k=1}^N a_k Y(z) z^{-k} + \sum_{k=0}^M b_k X(z) z^{-k} \quad (3.10)$$

Arranging Eq. 3.10, we obtain the transfer function $H(z)$ written in Eq. 3.1.2 and the frequency response of $H(z)$ is obtained by substituting $z = j\omega$ where ω denotes the frequency component. In this equation, the digital filter parameters are M , N , a_k and b_k .

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

$$H(z) = \frac{b_0 z^0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \quad (3.11)$$

Manipulating Eq. 3.11, we obtain the poles and zeros of the system in Eq. 3.13 depicted below.

$$H(z) = \frac{b_0 z^{-M}}{z^{-N}} \times \frac{z^M + \frac{b_1}{b_0} z^{M-1} + \frac{b_2}{b_0} z^{M-2} + \cdots + \frac{b_M}{b_0}}{z^N + a_1 z^{N-1} + a_2 z^{N-2} + \cdots + a_N} \quad (3.12)$$

$$H(z) = b_0 z^{-M+N} \times \frac{(z - z_1)(z - z_2) + \cdots + (z - z_M)}{(z - p_1)(z - p_2) + \cdots + (z - p_N)} \quad (3.13)$$

Re-arranging the Eq. 3.12 and 3.13, we arrive at Eq. 3.14. There G is called the system gain.

$$H(z) = G z^{-M+N} \times \frac{\prod_{k=1}^M (z - z_k)}{\prod_{k=1}^N (z - p_k)} \quad (3.14)$$

The gain, G in the previous equation is defined by the following equation:

$$G(\omega) = 20|H(e^{j\omega})| \quad (3.15)$$

The group delay is discussed next. It is defined as the negative of the rate change of phase with frequency. The quantity T has the time dimension that delays a signal by that time. This is the derivative of the phase response of the filter.

$$\tau(\omega) = -\frac{\delta\varphi(\omega)}{\delta\omega} \quad (3.16)$$

This measure gives the time distortion of a system with respect to phase and frequency.

3.1.3 Fourier Series

A periodic waveform can be expressed by the Fourier series (FS). The FS of a periodic waveform $f(t)$ is written by Eq. 3.17. In the equation, the first term a_0 is a constant. This is typically known as direct current (DC) or average component of $f(t)$.

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + b_n \sin(n\omega t) \quad (3.17)$$

Extending Eqs. 3.17–3.18, we find the amplitude of the fundamental frequency component or the first harmonic of the periodic waveform $f(t)$ as the summation of $\sqrt{a_1^2 + b_1^2}$ of the frequency ω , the amplitude of the second harmonic component of $f(t)$ is $\sqrt{a_2^2 + b_2^2}$ at 2ω and so on.

$$\begin{aligned} f(t) = & \frac{1}{2}a_0 + \\ & a_1 \cos(\omega t) + b_1 \sin(\omega t) + \\ & a_2 \cos(2\omega t) + b_2 \sin(2\omega t) + \\ & \dots + \\ & a_n \cos(n\omega t) + b_n \sin(n\omega t) \end{aligned} \quad (3.18)$$

3.1.4 Fourier Transform

The Fourier series analyses the periodic functions of time. Periodic signals generate line spectra with non-zero values only at specific frequencies. These are called harmonics. The non-periodic signals such as unit step, unit ramp, and rectangular signal are not well represented by a Fourier series, in such case we can apply Fourier transformation (FT). In Eq. 3.19, a non-periodic signal $f(t)$ has a period in the range of $-\infty$ to $+\infty$. The frequency ω is in radians. The $F(\infty)$ is the Fourier transform or the Fourier integral depicted below.

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \quad (3.19)$$

The Fourier transform is a complex function and it is represented in terms of real and imaginary forms or magnitude and phase forms as shown in Eq. 3.20.

$$F(\omega) = \operatorname{Re} F(\omega + j\operatorname{Im} F(\omega)) = |F(\omega)|e^{j\varphi(\omega)} \quad (3.20)$$

Extending Eq. 3.20, we write the Eq. 3.21:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \cos(\omega t) dt + j \int_{-\infty}^{+\infty} f(t) \sin(\omega t) dt \quad (3.21)$$

The properties of the Fourier Transform are not discussed here. However, its application in window filter design and spectral analysis is discussed briefly here. In Eq. 3.22, $z = e^{-j\omega}$ where $H(\omega)$ is the *FT* transform of $H(z)$. The $H(\omega)$ is also called the frequency response of the system.

$$H(\omega) = H(z) = \sum_{n=-\infty}^{+\infty} H(z)z^{-n} = \sum_{n=-\infty}^{+\infty} H(n)e^{-j\omega n} \quad (3.22)$$

In Eq. 3.23, $|H(\omega)|$ is the magnitude response and $\varphi(\omega)$ is the phase response:

$$H(\omega) = |H(\omega)|e^{-j\varphi(\omega)} \quad (3.23)$$

3.1.5 Discrete Fourier Transform and Fast Fourier Transform

The Discrete Fourier Transform (*DFT*) is a common tool that is used to establish a relationship between the time domain representation and the frequency domain representation. For example if $s[n]$ is a time domain signal at time instant, its frequency representation is $S(k)$ of $s[n]$ using *DFT* as shown in Eq. 3.24. There we considered that the signal is an N (finite) length signal. In the equation, the frequencies are $\frac{2\pi}{N}k$ for $k = 0, 1, 2, \dots, N - 1$. N represents the number of points that are equally spaced in the interval of $[0, 2\pi]$ on the unit circle in the z -plane. The frequency is:

$$S(k) = \sum_{n=0}^{N-1} s[n]e^{-j\frac{2\pi kn}{N}} \quad (3.24)$$

A particular algorithm to compute Discrete Fourier Transforms using dynamic programming is referred as Fast Fourier Transform (FFT).

The discrete Fourier transform (*DFT*) and *FFT* compute the signal amplitude spectrum and their power spectrum. The disadvantage of the *DFT* is that the underlying process is time-wise or space-wise invariant and it has insufficiency in time localization. This hinders extracting the information properly. This limitation is better managed by wavelet transformation.

3.1.6 Zero Padding

Since we deal with discrete samples, applying *FFT* requires knowing upfront the nature of the signal (e.g., its rate of change, duration, etc.). This knowledge is used then in selecting appropriate parameters for applying *FFT*; that is the *FFT* length. For computational efficiency reasons the length of *FFT* is selected as a power of 2. If the number of samples is a power of 2, then *FFT* can be used directly. Otherwise the length of the signal is increased to an N which is a power of 2, by adding additional zeros to the signal. For example if the original signal is $x(n)$, where $n = 0, 1, 2, \dots, N - 1$, and thus $\{x(0), x(1), x(2), \dots, x(N - 1)\}$ can be increased to $\{x(0), x(1), x(2), \dots, x(N - 1), 0, 0, 0, \dots, x(N^2)\}$. This is called zero padding. This does not change the statistical information of the signal, it is just a good representation for the continuous-frequency spectrum by increasing its resolution.

3.1.7 Overlap-Add and Overlap-Save Convolution Algorithms

Filter banks can be used to implement algorithms for the computation of convolutions. The two classic block processing schemes are the *overlap-add* and *overlap-save* algorithms for running convolution. In such case, a block of input is processed at a time typically with frequency domain circular convolution and the output is merged so as to achieve linear running convolution. Since the processing advances the computation which corresponds to downsampling the input by the step size, these schemes are multivariate in nature and have an immediate filter bank interpretation.

3.1.7.1 Overlap Add Method

In this scheme, suppose a filter is of the length L , the overlap-add algorithm takes a block of input samples length $M = N - L + 1$, and feeds it into a size N *FFT* ($N > L$), which applies the filter h . This results in a linear convolution of the signal with the filter. Since the size of the *FFT* is N , there will be $L - 1$ samples overlapping with adjacent blocks of size M , which are then added together, thus the name overlap-add. One can see that such a scheme can be implemented with an N -channel analysis filter bank downsampling by M , followed by multiplication or convolution in Fourier domain, upsampling by M and an N -channel synthesis filter bank.

The filters are used based on the short-time Fourier transform. In this method, a signal $x(n)$ is divided into M blocks of $x_m(n)$ and each block $x_m(n)$ is processed individually to generate $y_m(n)$. Here each block $x_m(n)$ is non-overlapping and the length of each block is L .

$$y_m(n) = x_m(n) * h(n) \quad (3.25)$$

3.1.7.2 Overlap Save Method

Given a length- L filter, the overlap-save algorithm performs the following: It takes N input samples, computes a circular convolution of which $N - L + 1$ samples are valid linear convolution outputs and $L - 1$ samples are wrap around effects. These last $L - 1$ samples are discarded. The $N - L + 1$ valid ones are kept and the algorithm moves up by $N - L + 1$ samples. The filter bank implementation is similar to the overlap-add scheme, except that analysis and synthesis filters are interchanged. The running convolution or point wise multiplication schemes used in the overlap-save and overlap-add algorithms can be improved by true convolution which results in a longer overall convolution if adequately chosen. Another possibility is to use analysis and synthesis filters based on fast convolution algorithms other than Fourier ones. In this method, the signal $x(n)$ is divided into M blocks of $x_m(n)$ and each block $x_m(n)$ is processed individually to generate $y_m(n)$. The same formulation used in overlap add method is applied for this method too. Here each block is overlapping except the first block.

3.1.8 Short Time Fourier Transform (STFT)

The signal is framed into m blocks by applying a window function of length N in such a way that each signal of the m blocks has N many samples where $m = 1, 2, \dots, M$ and $n = 0, 1, \dots, N - 1$. This segmented signal is called $s_m[n]$: this signal composes a frame. The processing of the framed signal by the windowing is shown in Eq. 3.26. The overlapping is normally one half or two third of the length of the window. The length of the window is equal to the signal length N . Thus the framed or the windowed signal $s_m[n]$ is the multiplication of the signal $s[n]$ by the window function $w[n]$. Multiplying the signal $s[n]$ by the window function $w[n]$, we attain the signal frames denoted by $s_m[n]$ for $m = 1, 2, \dots, M$ and $n = 0, 1, \dots, N - 1$. Shift r , which is the shift of samples between adjacent segments, has to be $r \leq N$. This is shown in the Eq. 3.26.

$$s_m[n] = s[n + rm]w[n] \quad (3.26)$$

3.1.8.1 Windowing and Window Functions

There are numerous different types of window functions available. Some commonly used window functions are the rectangular window function given in Eq. 3.27, the Hamming window function given in Eq. 3.29, and the Hanning window function given in Eq. 3.28 (Fig. 3.1).

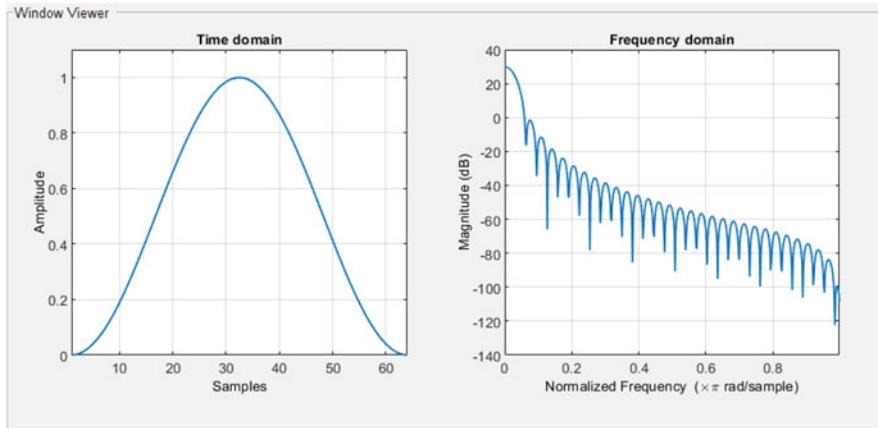


Fig. 3.1 The 64 point Hanning window

One purpose of the windowing is to improve the leakage problem and results in a spectral bias in the frequency analysis. The leakage problem arises when a signal is truncated into blocks. We will not discuss the derivations of the window functions and the constants used in equations. Some common window functions discussed below are commonly used in digital signal processing (*DSP*).

Rectangular Window The rectangular window is a type of B-spline function. It has also the names boxcar or Dirichlet window. It is a quite simple window, equivalent to replacing all but N values of a data sequence by zeros, making it appear as though the waveform suddenly turns on and off. The formula defining it is simple:

$$w[n] = 1 \quad \text{for} \quad 0 \leq n \leq N - 1 \quad (3.27)$$

Hanning Window The Hanning window, also known as the Hann window, is defined below with the Eq. 3.28:

$$w[n] = 0.5 - 0.5 \cos \left(\frac{2\pi n}{N - 1} \right) \quad \text{for} \quad n = 0, 1, 2, \dots, N - 1 \quad (3.28)$$

Hamming Window The Hamming Window Function is provided in Eq. 3.29 where $\alpha = 0.54$ and $\beta = 1 - \alpha = 0.46$. The window is optimized to minimize the maximum (nearest) side lobe. The height of this is about one-fifth of Hanning window written in Eq. 3.28.

$$w[n] = \alpha - \beta \cos \left(\frac{2\pi n}{N - 1} \right) = 0.54 - 0.46 \cos \left(\frac{2\pi n}{N - 1} \right) \quad (3.29)$$

The constants α and β are approximations of values $\alpha = 25/46$ and $\beta = 21/46$, which cancel the first sidelobe of the Hanning window by placing a zero at frequency $5\phi/(N - 1)$. Approximation of the constants to two decimal places substantially lowers the level of side lobes (Fig. 3.2).

Kaiser Window The Kaiser window is not a single function but rather a one-parameter family of window functions. It has a better control than the Hamming or Hanning window, but a use of the window also depends on the application. The α is the parameter which controls the ratio between the width of the main lobe and side lobe (Fig. 3.3).

I_0 is the zeroth order Bessel function.

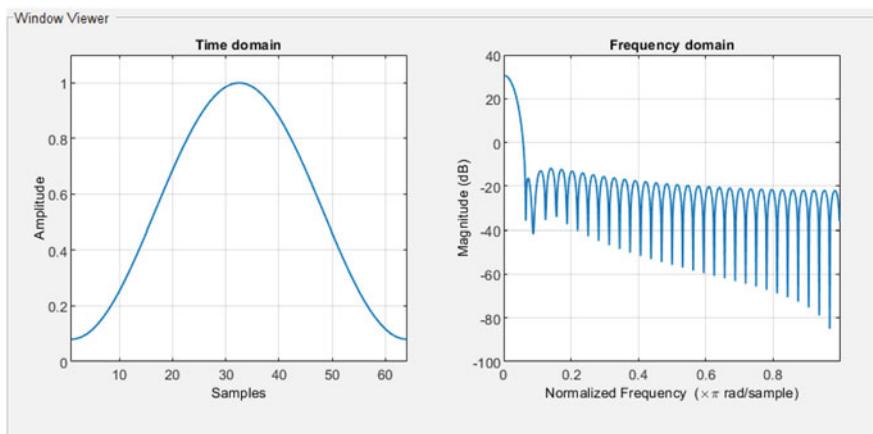


Fig. 3.2 The 64 point Hamming window

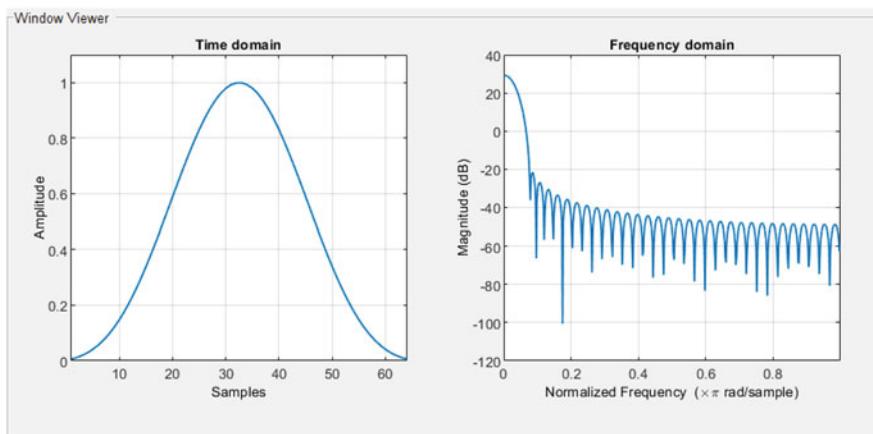


Fig. 3.3 The 64 point Kaiser window with parameter $\alpha = 7$

$$w[n] = \frac{I_0 \left(\pi \alpha \sqrt{1 - \left(\frac{2n}{N-1} - 1 \right)^2} \right)}{I_0(\pi \alpha)}, \quad 0 \leq n \leq N-1 \quad (3.30)$$

Chebyshev Window The Chebyshev window similarly to Kaiser window is not a single function but rather a one-parameter family of window functions. The parameter is easy to use as it requires specification of side lobe attenuation in terms of dB . The depicted example shows the 64-point filter with 100 dB attenuation (Fig. 3.4).

B-Spline Window B-spline windows can be obtained as k -fold convolutions of the rectangular window. They include the rectangular window itself ($k = 1$), the triangular window ($k = 2$) and the Parzen window ($k = 4$). Alternative definitions sample the appropriate normalized B -spline basis functions instead of convolving discrete-time windows. A k th order B -spline basis function is a piecewise polynomial function of degree $k-1$ that is obtained by k -fold self-convolution of the rectangular function.

3.1.8.2 Discrete Cosine Transform (DCT)

There exist real axis-only analogues of the DFT, namely the Discrete Cosine Transforms. A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT is a type of *FT* where only the \cos function is considered as its basis function. To obtain real, fast DCT algorithms one can mainly use a polynomial arithmetic technique or a matrix factorization technique. In DCT, computation is written as in the equation below.

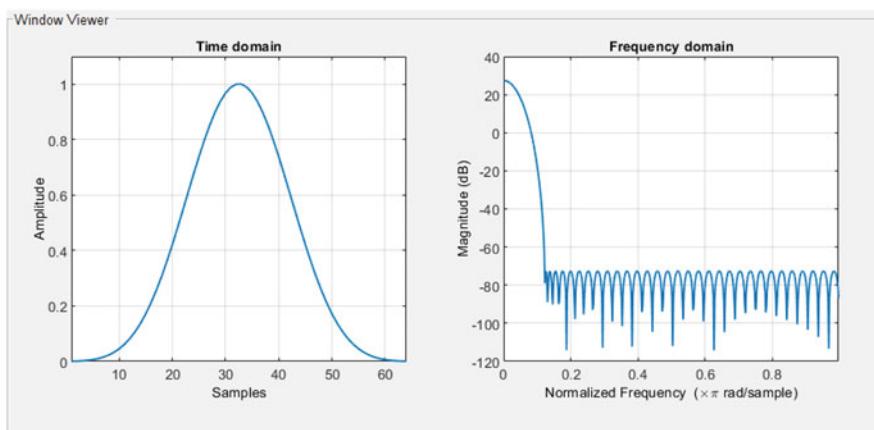


Fig. 3.4 The 64 point Chebyshev window with attenuation parameter 100

$$F(\omega) = \int_{-\delta}^{+\delta} f(t) \cos(\omega t) dt \quad (3.31)$$

The Discrete Cosine Transform (DCT) has a strong energy packing capability which is equivalent to [Karhunen-Loeve Transform \(KLT\)](#) in de-correlating signals (see the section on stochastic processes). De-correlation is sought to compact the signal information. This compact information is used for training and applying machine learning approach. The length N of the discrete cosine transform (DCT) of a signal $x(n)$ can be defined as:

$$C(k) = \alpha(k) \sum_{n=0}^{N-1} x[n] \cos \left[\frac{\pi(n + \frac{1}{2})k}{N} \right], \quad \forall \quad 0 \leq k \leq N - 1$$

where,

$$\alpha(k) = \sqrt{\frac{1}{N}}$$

Accordingly, the inverse DCT is given by

$$x(n) = \sum_{k=0}^{N-1} \alpha(k) C(k) \cos \left[\frac{\pi(n + \frac{1}{2})k}{N} \right], \quad \forall \quad 0 \leq n \leq N - 1$$

As mentioned the DCT is a real transform, that is it maps a real signal into a set of real DCT coefficients. We can define the DCT matrix C_N by

$$\{C(N)\}_{kn} = \alpha(k) \cos \left[\frac{\pi(n + \frac{1}{2})k}{N} \right]$$

The matrix of the DCT then becomes:

$$c = C_N x$$

$$x = C_N^T c$$

noting that, $C_N^{-1} = C_N^T$; This implies the matrix C_N is unitary.² Due to this fact, the Parseval's relation holds, that is, the energy of the signal is equal to the energy of the transform coefficients, such that:

² A complex square matrix A is unitary if its conjugate transpose A^* is also its inverse.

$$\sum_{n=0}^{N-1} C^2(n) = c^T c = (C_N x)^T C_N x = x^T C_N^T C_N x = x^T x = \sum_{n=0}^{N-1} x^2(n)$$

The DCT enjoys a very important property when it is applied to signals such as voice and video, most of the energy is concentrated in few DCT I transform coefficients $C(k)$. It can be seen that the energy is mostly concentrated in the first transform coefficients. Due to this property, the DCT is widely used in video compression schemes, because the coefficients with lowest energy can be discarded during transmission without introducing significant distortion in the original signal. In fact the DCT is part of most of the digital video broadcasting system in operation in the world. DCT is used in JPEG image compression, MPEG video compression.

There are eight different types of DCT. The most common DCTs are two dimensional. The DCT has four standard variants discussed in following paragraphs.

The DCTs are discretized solutions of the undamped harmonic oscillator equation together with certain homogeneous boundary conditions. First these functions are listed. There are four types, listed as DCT-I to DCT-IV. For a signal x of length N , and with δ_{kl} the (Kronecker) delta, the transforms are defined by:

- DCT-I

$$y[k] = \sqrt{\frac{2}{N-1}} \sum_{n=0}^{N-1} x(n) \frac{1}{\sqrt{1 + \delta_{n1} + \delta_{n(N-1)}}} \frac{1}{\sqrt{1 + \delta_{k1} + \delta_{k(N-1)}}} \cos\left(\frac{\pi}{N-1}(n)(k)\right)$$

- DCT-II

$$y[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \frac{1}{\sqrt{1 + \delta_{kl}}} \cos\left(\frac{\pi}{2N}(2n)(k)\right)$$

- DCT-III

$$y[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \frac{1}{\sqrt{1 + \delta_{n1}}} \cos\left(\frac{\pi}{2N}(n)(2k)\right)$$

- DCT-IV

$$y[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi}{4N}(2n)(2k)\right)$$

All variants of the DCT are unitary, hence to find their inverses, switch k and n in each definition is needed. DCT-I and DCT-IV are their own inverses. DCT-II and DCT-III are inverses of each other. The inverse DCT transforms such as IDCT-I, IDCT-II, IDCT-III and IDCT-IV are multiplied by DCT-I with $\frac{2}{n-1}$, DCT-III with $\frac{2}{n}$, and DCT-IV with $\frac{2}{n}$.

3.1.8.3 DCT I

With the proper edge point handled the DFT of the mirror data is typed DCT-I.

$$f_j = \frac{1}{2} \left(x_0 + (-1)^j x_{n-1} \right) \sum_{k=1}^{n-2} x_k \cos \left[\frac{\pi}{n-1} jk \right] \quad (3.32)$$

Thus an algorithm for DCT-I can be stated that executes recursively with DCT-I-IV algorithms.

DCT-I Algorithm: *Input*

$$n = 2^t (t \geq 1), n_1 = \frac{n}{2}, \mathbf{x} \in \mathbb{R}^{n+1}$$

1. If $n = 2$ then

$$= \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & \sqrt{2} & 0 \\ 1 & 0 & -1 \end{bmatrix} \mathbf{x}$$
2. If $n \geq 4$ then

$$[u_j]_{j=0}^n \equiv H_{n+1} \mathbf{x}$$

$$z_1 = \cos_1 \left([u_j]_{j=0}^{n_1}, n_1 + 1 \right) \quad z_2 = \cos_3 \left([u_j]_{j=n_1+1}^n, n_1 \right)$$

$$y = P_n T + 1 Z_1 T, Z_2^T$$

3.1.8.4 DCT-II

The DCT-II matrix is orthogonal and decomposes using the following formula:

$$f_j = \sum_{k=0}^{n-1} x_k \cos \left[\frac{\pi}{n} \left(k + \frac{1}{2} \right) \right] \quad (3.33)$$

The computation is carried out by applying an algorithm:
DCT-II Algorithm: Input: $n = 2^t (t \geq 1), n_1 = \frac{n}{2}, (x) \in \mathbb{R}^n$.

1. If $n = 2$, then

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{x}$$

2. If $n \geq 2$ then

$$[u_j]_{j=0}^n \equiv H_n \mathbf{x}$$

$$z_1 = \cos_2 \left([u_j]_{j=0}^{n_1-1}, n_1 \right) \quad z_2 = \cos_4 \left([u_j]_{j=n_1}^{n-1}, n_1 \right)$$

$$y = P_n^T Z_1^T, Z_2^T$$

3.1.8.5 DCT III

The $DCT - III$ it generates an orthogonal transformation but outputs a real-even half shifted DFT .

$$f_j = \frac{1}{2} \sum_{k=0}^{n-1} x_k \cos \left[\frac{\pi}{n} \left(k + \frac{1}{2} \right) \right] \quad (3.34)$$

By using the well known transpose property between $DCT - II$ and $DCT - III$ we can state an algorithm for $DCT - III$. This algorithm executes recursively with the $DCT - II$ and $DCT - IV$ algorithms:

$DCT - III$ Algorithm: Input: $n = 2^t$ ($t \geq 1$), $n_1 = \frac{n}{2}$, $(x) \in \mathbb{R}^n$.

1. If $n = 2$, then

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{x}$$

2. If $n \geq 2$ then

$$[u_j]_{j=0}^n \equiv H_n \mathbf{x}$$

$$z_1 = \cos_3 \left([u_j]_{j=0}^{n_1-1}, n_1 \right) \quad z_2 = \cos_4 \left([u_j]_{j=n_1}^{n-1}, n_1 \right)$$

$$y = H_n^T Z_1^T, Z_2^T$$

3.1.8.6 DCT IV

The $DCT - IV$ generates an orthogonal matrix. There are several forms of algorithmic realizations. Many sophisticated sub-band coding and filters are based on this transformation.

$$f_j = \sum_{k=0}^{n-1} x_k \cos \left[\frac{\pi}{n} \left(j + \frac{1}{2} \right) \left(k + \frac{1}{2} \right) \right] \quad (3.35)$$

The $DCT - IV$ matrix becomes orthogonal (and thus, being clearly symmetric, it is its own inverse) if one further multiplies by an overall scale factor of $\sqrt{\frac{2}{N}}$.

$$[C_{N+1}^I]_{kn} = \sqrt{\frac{2}{N}} \left[\epsilon_k \epsilon_n \cos \left(\frac{\pi n k}{N} \right) \right]; \quad k, n = 0, 1, 2, \dots, N$$

$$[C_{N+1}^I]^{-1} = [C_{N+1}^I]^T = C_{N+1}^I$$

$$[C_{N+1}^{II}]_{kn} = \sqrt{\frac{2}{N}} \left[\epsilon_k \cos \left(\frac{\pi(2n+1)k}{2N} \right) \right]; \quad k, n = 0, 1, 2, \dots, N$$

$$[C_N^{II}]^{-1} = [C_N^{II}]^T = C_N^{II}$$

$$[C_{N+1}^{III}]_{kn} = \sqrt{\frac{2}{N}} \left[\epsilon_k \cos \left(\frac{\pi(2k+1)n}{2N} \right) \right]; \quad k, n = 0, 1, 2, \dots, N$$

$$[C_N^{III}]^{-1} = [C_N^{III}]^T = C_N^{III}$$

$$[C_{N+1}^{IV}]_{kn} = \sqrt{\frac{2}{N}} \left[\epsilon_k \cos \left(\frac{\pi(2k+1)(2n+1)}{4N} \right) \right]; \quad k, n = 0, 1, 2, \dots, N$$

$$[C_N^{IV}]^{-1} = [C_N^{IV}]^T = C_N^{IV}$$

Where

$$\epsilon_p = \begin{cases} \frac{1}{\sqrt{2}} & \forall p=0 \quad or \quad p=N \\ 1 & Otherwise \end{cases}$$

A variant of the *DCT - IV*, where data from different transforms are overlapped, is called the modified discrete cosine transform (*MDCT*).

A question naturally arises is which is the transform that maximizes this energy concentration. Given a statistical distribution of an ensemble of signals, the optimum transform in terms of the energy compaction capability is the Karhun-Loeve transform (*KLT*). It is defined as the transform that diagonalizes the autocovariance matrix of a discrete random process. There is a different type of *KLT* for each of the different signal statistics. However, it can be shown that the *DCT* approximates the *KLT* when the signals can be modeled as Gauss-Markov processes with correlation coefficients near to 1. This is reasonable a good model for several useful signals, like video for instance. For this signals, the *DCT* is indeed approximately optimum in terms of energy compaction capability, where it is widely used.

3.1.9 Wavelet Transform

Signals are analyzed with respect to time and space by the wavelet transforms. This decomposes signal into set of wavelet basis functions. These wavelets are translated with the time or the frequency aspect of the signal.

If a discrete real data sequence is $y_i, \forall i = 1, 2, \dots, n$ where $n = 2^j \forall j \geq 0$ such as:

$$y = (y_1, y_2, y_3, \dots, y_n) \quad (3.36)$$

The detailed information is extracted in sequence at different scales and at different locations at multiscale information from the vector y . The detail information is extracted as:

$$d_k = y_{2k} + y_{2k-1} \forall k = 1, 2, 3, \dots, \frac{n}{2} \quad (3.37)$$

where

$$d_1 = y_2 - y_1$$

$$d_2 = y_4 - y_3$$

...

Each of the d_k reveals information at and around the points of y_{2k} and its immediate neighbor.

Now the information at coarser scales is obtained by applying Eq. 3.38:

$$c_k = y_{2k} + y_{2k-1} \quad \forall k = 1, 2, 3, \dots, \frac{n}{2} \quad (3.38)$$

Now $c_{k=1}^{\frac{n}{2}}$ is a set of scaled local averages and the information in c_k is a coarsening of that in the original y vector.

Each c_k contains information originating from both y_{2k} and y_{2k-1} .

Question 3.1 Explain the multi-scale situation for the number sequence:

$$y = (y_1, y_2, y_3, \dots, y_n) = (1.1.7.9.2.8.8.6).$$

Solution:

$n = 8$ and $j = 3$ as $2^3 = 8$.

$$d_{2,1} = y_2 - y_1 = 1 - 1 = 0$$

for the remaining d coefficients, at level $j = 2$ and

$$\begin{aligned}
 d_{2,2} &= y_4 - y_3 & = 9 - 7 = 2 \\
 d_{2,3} &= y_6 - y_5 & = 8 - 2 = 6 \\
 d_{2,4} &= y_8 - y_7 & = 6 - 8 = -2 \\
 2^{j-1} &= \frac{n}{2} & = 4
 \end{aligned}$$

The local average information is obtained by addition operations. That is:

$$\begin{aligned}
 c_{2,1} &= y_2 + y_1 & = 1 + 1 = 2 \\
 c_{2,2} &= y_4 + y_3 & = 9 + 7 = 16 \\
 c_{2,3} &= y_6 + y_5 & = 8 + 2 = 10 \\
 c_{2,4} &= y_8 + y_7 & = 6 + 8 = 8
 \end{aligned}$$

Since there are 8 input elements, that is $y_i \forall i = 1, 2, 3, \dots, 8$ in y generates 4 d_2 and 4 c_2 elements. Here $d_{j,k}$ are called wavelet coefficients and $c_{j,k}$ are called father wavelet or scaling function coefficients. For example, the final 42 indicates that the sum of the whole original sequence is 42. The 18 indicates that the sum of the last quarter of the data minus the third quarter is four.

The inverse wavelet can be expressed by equation below

$$c_{j-1,2k} = \frac{(c_{j-2,k} + d_{j-2,k})}{2} \quad (3.39)$$

The wavelet coefficients are sparse meaning that the wavelets are piecewise sparse smooth functions. The sparsity is a consequence of the unconditional basis wavelet property. The input sequence can be thought to possess an energy or norm as defined by:

$$\|y\|^2 = \sum_{i=1}^8 y_i^2 \quad (3.40)$$

Here the norm of the input sequence is $1 + 1 + 1 + 49 + 81 + 4 + 64 + 64 + 36 = 296$. The transform coefficients from finest to coarsest are:

$$(0, 2, 6, -2, 14, 4, 6, 42)$$

The norm of the wavelet coefficients is:

$$0^2 + 2^2 + 6^2 - 2^2 + 14^2 + 4^2 + 6^2 + 42^2 = 2056$$

The norm of energy of the output sequence is much larger than that of the input.

3.1.9.1 Multiresolution Analysis (MRA)

The *MRA* is a nested sequences of linear spaces formulated based on orthonormal and compactly supported wavelet bases. The spaces are a set of functions regulated on some principle theory.

In multiresolution analysis, a vector of data produced a set of detail coefficients and a set of smooth coefficients by differencing and averaging in pairs. It can be appreciated that a function that has reasonable non-zero fine-scale coefficients are very small or zero. A low-resolution function is progressively revealed to finer details by inventing a new layer of detailed coefficients and working back to the sequence that would have produced the original.

Multiresolution analysis leads us on to scale spaces of functions. The space V_j containing functions with details up to some finest scale of resolution. These spaces contain functions with less detail. The larger j would indicate V_j containing functions with finer and finer scales. If a function was in V_j , then it must also be in $V_l \forall l > j$ such that $V_j \subset V_l$ i.e.

$$\cdots CV_{-2}CV_{-1}CV_0CV_1CV_2CV_3\cdots$$

As j becomes large and positive we include more and more functions of increasingly finer resolution. Eventually, as j tends to infinity we want to include all functions: mathematically this means that the union of all the V_j spaces is equivalent to the whole function space we are interested in. As j becomes large and negative, we include fewer and fewer functions, and detail is progressively lost. As j tends to negative infinity the intersection of all the spaces is just the zero function.

3.1.9.2 Wavelet Packet Transform

The wavelets are some little waves like functions which regulate the signal in the time and frequency scales. The wavelets are objects that oscillate but decay fast. Thus wavelets are little. In equation presented below a wavelet is composed with a combination of dilation and translation.

$$\varphi_{j,k}(x) = 2^{\frac{j}{2}}\varphi(2^j x - k) \quad \forall i, j \in \mathbb{Z} \quad (3.41)$$

In Eq. 3.41, $\varphi_{j,k}(x)$ is an orthonormal basis for $L^2(R)$ where R denotes a real number and \mathbb{Z} denotes an integer. A function $f(x)$ is decomposed into an orthonormal basis of, $\varphi_{j,k}(x)$ shown in Eq. 3.41. In next equation, $d_{j,k}$ is wavelet coefficients of $f(x)$.

$$f(x) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} d_{j,k} \varphi_{j,k}(x) \quad (3.42)$$

The $d_{j,k}$ can be obtained by the inner product of $\langle f, \varphi_{j,k} \rangle$ where

$$d_{j,k} \varphi_{j,k}(x) = \int_{-\infty}^{+\infty} f(x) \varphi_{j,k}(x) dx \quad (3.43)$$

3.1.9.3 Best Basis

A functional space is a collection of functions that satisfies a certain mathematical structural pattern. The finite energy space $L^2(-\infty, +\infty)$ is a collection of functions that are square integrable, such as:

$$\int_{-\infty}^{+\infty} |f(x)|^2(x) dx < \infty$$

The wavelet analysis deals with the expansion of functions in terms of a set of basis functions.

3.1.10 Windowing Signal and the DCT Transforms

One starts with an interval and splits the given interval into two overlapping intervals and construct a basis for each of these. The integration is obtained by a so-called rising cut-off function. Then the local cosine transform in the spectral analysis reduces the blocking artifacts and it smooths the signal. The overlapping between the adjacent blocks is used in the local cosine trigonometric transformation. Hence windowing is in principle an extension of segmentation.

The window function determines the weights when sampling the signals. It says with which weight the sampled values enter:

- The starting point are segments that are half-open intervals $I_j = [a_j; a_{j+1})$.
- Next a number r is chosen such that $a_j + r < a_{j+1} - r, \forall j$.
- The intervals are refined to $[a_j - r; a_j + r)$.

Using these intervals one obtains a segmentation. This has still the disadvantage that one has discontinuities for description at the boundaries of the intervals. The number r describes the overlapping. This is used for smoothly combining the intervals. The understanding of overlapping will be our next task. A major aspect now is to smooth the intervals. For this we use the overlapping. The smooth windowed signal is obtained by applying so-called trigonometric cutoff- functions.

For this we begin by defining a function $\beta(t)$.

- The function $\beta(t)$ is such that β is a function from the arguments given a real interval, i.e. $\beta(t) \in R^d$ with $0 \leq d$ satisfying the following condition:

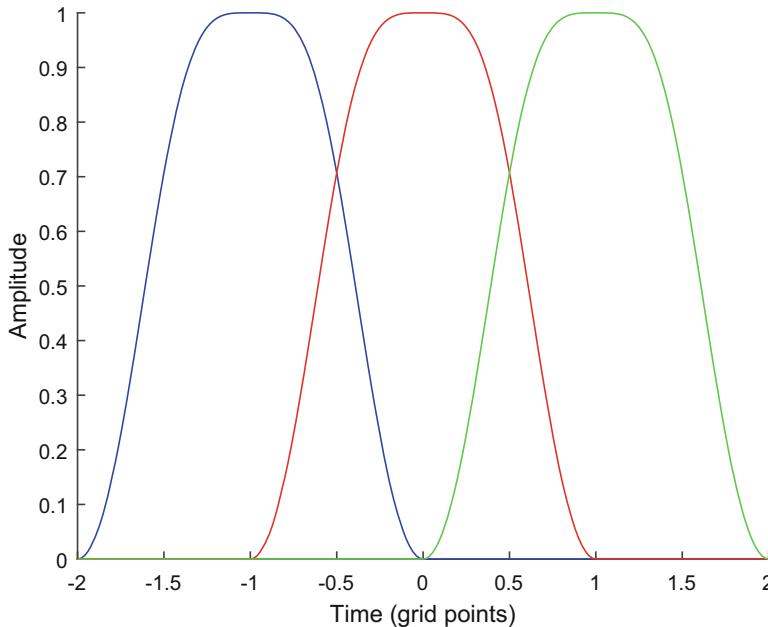


Fig. 3.5 Rising cut-off function

This condition will now be extended to a full definition in the interval $-1, 1$. We get it as a continuous function:

In order to obtain smoothness the introduced constants need to satisfy some conditions. To ensure smoothness we take a number $\epsilon > 0$ such that $a_j + 1 - a_j \geq \epsilon$ for each $j \in \mathbb{Z}$. Figure 3.5 shows an example of smooth integration using the rising cut-off functions.

3.2 Analysis and Comparison of Transformations

In the Fourier transform, the basic functions are the sinusoidal functions, where the DCT is also a type of Fourier transform but it uses cosine functions as the basis function. In wavelet transform, the wavelet is the basis function. The *FFT* is represented in terms of *sin-e* and *cos-ine* functions, the *DCT* are represented in terms of *cos* functions. The wavelets are represented in terms of shifted and scaled version of mother wavelet. In wavelet transformation, the time and frequency information can be revealed by the basic functions. The Fourier basis represents only the frequency domain or the frequency information well. The Fast Fourier transform (*FFT*) uses complex numbers. The *DCT* uses only real numbers. The *FFT* is twice the length of the *DCT*. A most common uses of *DCT* is compression.

DCT generates decorrelated features in image processing and image compression feature extraction stages which can be effectively used in Gaussian distribution with a diagonal covariance matrix. The cosine transformation is the even function of the *FFT*. The *DCT* is preferable in compression because it generates continuous extension at the boundaries. Generally, the Fourier series has discontinuities which reduce the rate of convergence of the function. The *DFT* generates discontinuities at both boundaries of the signal. This is in contrast with *DCT* which yields continuous extension at the boundaries at both sides of the signal. The smoother the function, the fewer terms of *DFT* or *DCT* are required to represent the signal accurately, and hence it yields the signal that is more compressed. This is the reason why *DCT* is preferred in signal compression. The *DCT* and wavelet transforms are most commonly used in speech coding, speech compression, image compression, image de-noising, video signal processing, time frequency analysis of the signals etc.

3.3 Background Information

Signals are some abstract or physical or symbolic information. Revealing information that lies in the signal requires processing the signal. Such processings are managed through some transformations. These transformations are mathematical, statistical, logical, and/or algorithmic. These transformations can be based on simple or complex mathematical approaches. In this chapter we have introduced the Laplace transform, z-transform, FFT, DFT, DCT, and wavelet transform. The transformations can be standard or application dependent but the objective is to transform the information from one form to another form.

The signal processing origin can be found in the classical numerical analysis techniques of the seventeenth century. The digital refinement of these techniques can be found in the digital control systems of the 1940s and 1950s. The history records that applying some techniques in numerical analysis and digitization, required in turn the conversion of the signal from analog to digital. This conversion, on the other hand, required development of theories of transforming analog signal into discrete form. This led to development of theories and understanding of conditions when such transformation can be applied (Shannon Theorem).

This development required understanding and contrasting the Laplace transformation (continuous domain transformation named after named after Pierre-Simon Laplace, a French mathematician and astronomer) with Z-transformation (discrete domain). The basic idea of z-transform was also introduced by Pierre-Simon Laplace, but the z-transformation had gone through several modifications to come to turn into todays' z-transformation. Those techniques are used for filter design or system analysis in signal processing, control system, medical science, psychology, economics and finance and in many other areas for system design.

Similarly Fourier transform is also named after French mathematician and physicist Joseph Fourier. This is the standard and most commonly used transform for spectrum analysis. Then there is short time Fourier transform (*STFT*) to broaden

the scope of *FFT*, but *STFT* uses different type of window functions to transform the signal.

Then there is wavelet transform for multiresolution analysis which performs better than the *STFT*, but the basic objective remains the same. The wavelet concept was introduced by Yves Meyer (1992), a French mathematician and engineer. A slightly modified version of wavelet transform is *LLT* introduced by French mathematicians Eva Wesfried and Mladen Victor Wickerhauser.

More about linear time invariant systems, their transformation and more advanced transformation such as wavelet, *LLT* can be found from the list mentioned in references.

References such as Rao (1990), Keller (2004), Addison (2002), Strang (1993), Byrne (2013), Smith (2011), Wesfried (1993), Nason (2008), Diniz (2001), and Selesnick (2007) are recommended for more information.

3.4 Exercises

Question 3.2 Express the Laplace transform for the following trapezoidal pulse presented in the Fig. 3.6:

Question 3.3 Find the *DFT* of the output of the causal system provided with difference equation below:

$$y[n] = -x[n] + 2x[n - 2] - x[n - 4]$$

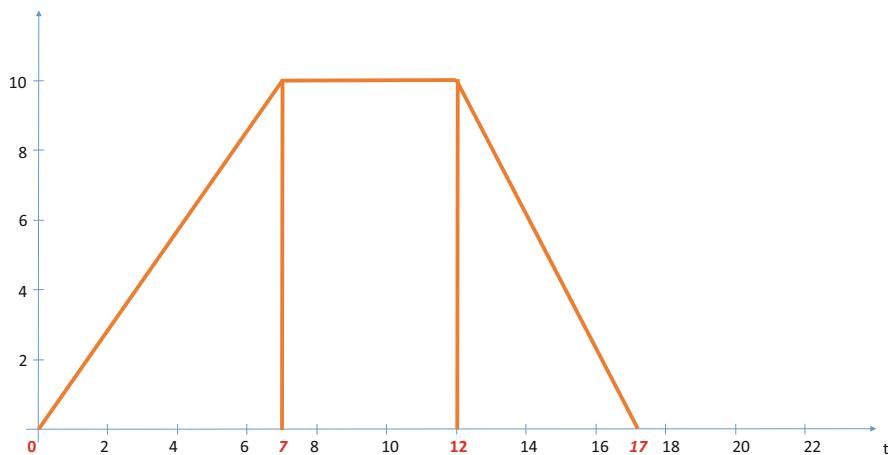


Fig. 3.6 Trapezoidal signal

Find the transfer function, impulse response, and frequency response of the system. Is the system linear? Is the system stable?

Question 3.4 The finite length input signal is given by:

$$\begin{aligned}x[n] &= [12 - 3 - 2 - 2] \\&= \delta[n] + 2\delta[n - 1] - 3\delta[n - 2] - 2\delta[n - 3] - \delta[n - 4], \quad \forall 0 \leq n \leq 4\end{aligned}$$

Draw the graph of $x[n]$ of the system. Draw the pole zero diagram of the system.

Question 3.5 A 5 kHz sinusoidal signal is sampled at 40 kHz and 128 samples are collected and used to compute 128 samples. What is the time duration in seconds of the collected samples? At what *DFT* indices do we see any peaks?

References

- [Rao1990] K. R. Rao and P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications, Academic Press, Boston MA, 1990
- [Keller2004] Keller, Wolfgang. Mathematik: Wavelets in Geodesy and Geodynamics: Walter de Gruyter, 2004
- [Addison2002] Addison, Paul S. The Illustrated Wavelet Transform Handbook, IOP Publishing Ltd, UK, 2002
- [Strang1993] Strang, Gilbert. Wavelet Transforms Versus Fourier Transforms, American Mathematical Society, Vol 28, Number 2, April, 1993
- [Byrne2013] Charles L. Byrne, Mathematics of Signal Processing: A First Course, MIT Press, 2013
- [Smith2011] Smith, Steven W. The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, 2011
- [Meyer1992] Meyer, Y., Wavelets and Operators. Cambridge: Cambridge University Press. ISBN 0-521-42000-8., 1992
- [Wesfried1993] Eva Wesfried, E. and Mladen Victor Wickerhauser, M. V. , Adapted Local Trigonometric Transforms and Speech Processing, IEEE Transactions on Signal Processing 41(12), October 1993
- [Nason2008] Nason, G., Wavelet Methods In Statistics With R, Springer, August 2008
- [Diniz2001] Diniz. P S. R. Digital Signal Processing: System Analysis and Design, Prentice Hall, NJ, USA, Cambridge University Press, 2001
- [Selesnick2007] Selesnick , I., Wavelets, a modern tool for signal processing, Physics Today. 60(10):78–79, October 2007

Chapter 4

Digital Filters



Overview

One of the main purposes of digital filtering is to improve the quality of the signal. In this chapter, we give an overview of digital filtering. This often uses transformations in order to maintain the desired information in the presence of undesired signals that could corrupt it. Finite Impulse Response (*FIR*) and Infinite Impulse Response (*IIR*) filters are linear time invariant filters because the coefficients of the filters do not vary with time (i.e., they are time invariant) as well as because the signal is decomposed into a linear combination of basic signals (i.e., the linearity property). The *FIR* filter is non-recursive as this does not have feedback. In contrast the *IIR* filter has a recursive component because of feedback. The application of *FIR* filters to signal processing is more computationally involved than with *IIR* filters because *FIR* needs more coefficients than *IIR* filters. A better understanding of digital filters is obtained through study of its workings in the frequency domain rather than in the time domain. The filter implementation is typically a convolution of the time domain impulse response and the sampled signal. A filter is designed with a frequency domain impulse response which is as close to the desired ideal response as can be generated given the constraints of the implementation. The frequency domain impulse response is then transformed into a time domain impulse response to be converted to the coefficients of the filter. This chapter will discuss more on this.

4.1 Introduction

Signals in nature are defined by their continuously varying values. The resulting analog (continuous) waveform is typically denoted by $x(t)$. On the other hand, discrete-time (*DT*) signals are, as the name implies, values that exist only at specific

and discrete instances of time. Discrete Time Signals are functions (real or complex-valued) of an integer-valued independent variable, n , which is called “sample index” or “Discrete Time index”.

The resulting discrete time signals are denoted by $x(n)$. A signal is indicated as a lower case letter:

$$\begin{aligned} x &\Rightarrow \text{Signal Name} \\ x(n) &\Rightarrow n\text{'th sample of Discrete Time Signal} \end{aligned}$$

The studied systems process signals. The working of a system is best described through its ability to filter signals. In discrete domain filters are termed digital filters. The filtering operation is depicted by its desired responses: Low-Pass, Band-Pass, High-Pass, Band-Stop, etc. An example of Low-Pass filter is presented in the Fig. 4.1:

There are two basic types of digital filters: **Finite Impulse Response digital filter (FIR)** and **Infinite Impulse Response digital filter (IIR)** filters. The general form of the digital filter is the difference Eq. 4.1 where $y(n)$ is the current filter output, the $y(n - i)$ is the previous filter outputs, the $x(n - i)$ are current or previous filter inputs, the a_i 's are the filter's feed forward coefficients define the zeros of the filter, the b_i 's are the filter's feedback coefficients that define the poles of the filter, and the largest of N and M defines the filter's order. *IIR* filters have one or more non zero feedback coefficients. As a result of the feedback term if the filter has one or more poles, once the filter has been excited with an impulse there is always an output.

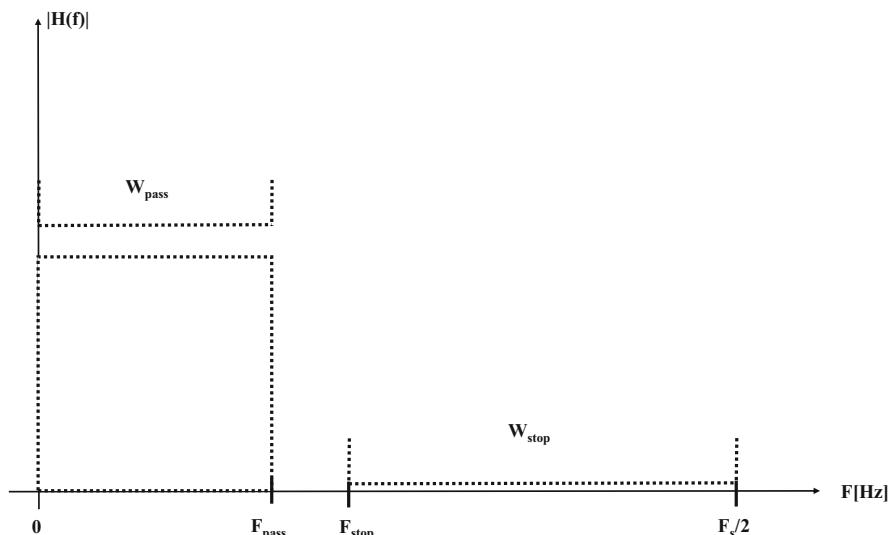


Fig. 4.1 The figure shows an ideal low-pass filter specification. The F_{pass} depicts pass-band, F_s depicts the sampling period, while the other notation is self-explanatory

The *FIR* filter is shown in Eq. 4.1 below:

$$y(n) = \sum_{j=0}^M b_j x(n-j) = b_0 x(n) + b_1 x(n-1) + \cdots + b_N x(n-M) \quad (4.1)$$

The *IIR* filter is shown in Eq. 4.2.

$$y(n) = \sum_{i=1}^N a_i y(n-i) + \sum_{j=0}^M b_j x(n-j) \quad (4.2)$$

The *FIR* and *IIR* filters can be generated by applying the Fourier transformation, a windowing method, and they can be optimized to achieve a specific desired goal. This chapter will not discuss these methods, but rather it discusses application of *FIR* and *IIR* filters to real world signals.

4.1.1 *FIR* and *IIR* Filters

The *FIR* filter has a linear phase response and all zero structure when the filter's coefficients are symmetric, as is the case in most standard filtering applications. A *FIR*'s implementation noise characteristics are easy to model, especially if no intermediate truncation is used. An *FIR* filter is stable and used in architectures in decimating or interpolating applications. An *IIR* filter's poles may be close to or outside the unit circle in the z plane. This may result in the *IIR* filter stability issues, especially after quantization is applied. In *IIR* filters, as depicted in Eq. 4.2, b_i are non-recursive filter coefficients, or feed forward filter coefficients, and a_i are the recursive or feedback filter coefficients. The poles which are the coefficients of the denominator of Eq. 4.2 have to be inside the unit circle in the z -domain for a stable *IIR* filter and the zeroes which are the coefficients of the polynomials of Eq. 4.1 have to be inside the unit circle in the z -domain for the stable *FIR* filter. The *FIR* filter has a linear phase, but the linear phase is not often reachable in the *IIR* filter.

Typical *FIR* and *IIR* filters are: low pass, high pass, band pass and band stop. We will also show their application in different types of filters, also explain why these filters are not sufficient for analysing the signals and why it is needed to adopt other filtering approaches. In low pass filter design, the poles lie near the unit circle corresponding to low frequencies where frequency $\omega = 0$ and the zeroes are placed close to high frequencies. In high pass filter designing, the poles lie close to high frequencies and zeroes lie close to low frequencies. In the band pass filter, one or more pair of poles are complex conjugates. The filter design specification includes the location of passbands and stop bands, the minimum stopband attenuation, the maximum passband ripple, filter orders, and the shape of the response in the filter bands. A typical *IIR* filter response starts with the prototype of an analog filter. Then

an S -domain to Z -domain transformation is used to generate a set of digital filter coefficients. Common methods of designing *FIR* filter responses are windowing and frequency sampling. There exists some standard commercially available digital filter design software packages such as in Matlab, Octave, and Scilab. In Eq. 4.1, b_i denotes the *FIR* filter coefficients, $M + 1$ is *FIR* filter length. This filter, as all of *FIR* filters, is non-recursive. By applying Z transform to the Eq. 4.1 we get:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^M b_i z^{-i} \quad (4.3)$$

Similarly the general architecture of the *IIR* filter can be formulated by Eq. 4.4.

$$y(n) = \sum_{i=1}^N a_i y(n-i) + \sum_{j=0}^M b_j x(n-j)$$

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + \cdots + a_N y(n-N) + \\ b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \cdots + b_M x(n-M) \quad (4.4)$$

Some basic characteristics of *FIR* and *IIR* filters are:

1. *FIR* filter has linear phase characteristics and *IIR* has non-linear phase characteristics.
2. The order of the *FIR* filter is generally high, on the other hand, the *IIR* filter order is usually low.
3. The *FIR* filter is stable, where a *IIR* filter can be unstable.

The z -transform of Eq. 4.4 is shown in Eq. 4.5.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{j=1}^N a_i z^{-j}} = \frac{B(z)}{A(z)} \quad (4.5)$$

Typically designing various filters requires understanding the procedure of designing the low-pass filter. The relationship between the various filters is provided below.

- Low-Pass Filter

$$h(n) = \begin{cases} \frac{\Omega_c}{\pi} n = \frac{\sin(\Omega_c n)}{\pi n} & \& -M \leq n \leq M \\ 0 & n = 0 \end{cases} \quad (4.6)$$

- High-Pass Filter

$$h(n) = \begin{cases} \frac{\pi - \Omega_c}{\pi} & n = \frac{\sin(\Omega_c n)}{\pi n} \quad \& -M \leq n \leq M \\ 0 & n = 0 \end{cases} \quad (4.7)$$

- Band-Pass Filter

$$h(n) = \begin{cases} \frac{\Omega_h - \Omega_l}{\pi} & n = \frac{\sin \sin(\Omega_h n)}{\pi n} - \frac{\sin \sin(\Omega_l n)}{\pi n} \quad \& -M \leq n \leq M \\ 0 & n = 0 \end{cases} \quad (4.8)$$

$$h(n) = \begin{cases} \frac{\pi - \Omega_h + \Omega_l}{\pi} & n = \frac{\sin \sin(\Omega_h n)}{\pi n} + \frac{\sin \sin(\Omega_l n)}{\pi n} \quad \& -M \leq n \leq M \\ 0 & n = 0 \end{cases} \quad (4.9)$$

The FIR filter is linear phased and this linearity can be of four types:

- **Type I:** Its impulse response is symmetrical and the length of the impulse response is odd.
- **Type II:** Its impulse response is asymmetrical and the length of the impulse response is even.
- **Type III:** Its impulse response is asymmetrical and the length of the impulse response is odd.
- **Type IV:** Its impulse response is asymmetrical and the length of the impulse response is even.

A simple low or high pass filter by itself is not good enough for processing the complicated signals such as speech, or image. But they are most often used in some preliminary type of signal processing. For instance, a simple low pass filter such as first order low pass filter is commonly used in anti-aliasing. On the other hand, a common use of the simple high pass is for pre-emphasizing the speech signal.

Question 4.1 Design a type I low-pass filter according to specifications provided below:

- pass-band frequency, $\omega_p = 0.2\pi$
- stop-band frequency, $\omega_s = 0.3\pi$
- pass-band tolerance, $\delta_1 = 0.1$
- stop-band tolerance, $\delta_2 = 0.01$

Answer:

The first step is to select a suitable filter order. From the requirements we use the peak error specification: $\delta_2 = 0.01$, to obtain the requirement of $20 \log_{10}(\delta_s) = -40\text{dB}$. In addition we use main-lobe width as $\omega_s - \omega_p = 0.3\pi - 0.2\pi = 0.1\pi$. It can be noted that this transition width relates to number of points used for filter (e.g., M) with the following $0.1\pi = \frac{8\pi}{M}$. Hence, filter length $M \geq 80$ and filter order $N \geq 79$.

Since, Type I filters need to have an even order, $N = 80$. The second step is to specify the ideal response. The band-edge of the frequency of the ideal response

filter is the midpoint between ω_s and ω_p , that is $\omega_c = \frac{(\omega_s + \omega_p)}{2} = \frac{(0.2\pi + 0.3\pi)}{2} = 0.35\pi$

$$H_d(\omega) = \begin{cases} 1 & \text{if } |\omega| \leq 0.25\pi \\ 0 & \text{if } 0.25\pi < |\omega| < \pi \end{cases}$$

The third step is to compute the coefficients of the ideal filter. The ideal filter coefficients h_d gives by the inverse discrete time Fourier transform of $H(\omega)$.

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H_d(\omega) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{+\omega_c} e^{j\omega n} d\omega = \frac{\omega_c}{\pi} \frac{\sin(\omega_c n)}{\omega_c n}$$

In order to make this system be causal we must delay its impulse response by $\frac{N}{2} = \frac{80}{2} = 40$.

Coefficients of the ideal filter are:

$$h(n) = \frac{\sin\{0.5\pi(n - 40)\}}{\pi(n - 40)}$$

Finally we can compute the coefficients of the ideal filter. Given the value of the N , the N point inverse Fast Fourier Transform is computed with the frequency spacing $\frac{2\pi}{N}$ rad/sample:

$$H_d(p) = H_d\left(\frac{2\pi p}{N}\right), \quad p = 0, 1, \dots, N - 1 \quad (4.10)$$

If the inverse FFT, hence the filter coefficients, are to be purely real valued, the frequency response must be conjugate symmetric:

$$H_d\left(\frac{-2\pi p}{N}\right) = H_d^*\left(\frac{2\pi p}{N}\right) \quad (4.11)$$

From Eqs. 4.10 and 4.11 we find:

$$H_d(N - p) = H_d^*(p) \quad \text{for } p = 1, 2, \dots, \left(\frac{N}{2} - 1\right)$$

The inverse FFT of $H_d^*(p)$ is an N -sample time domain function $h'(n)$. For $h'(n)$ to be an accurate approximation of $h(n)$, N must be made large enough to avoid time-domain aliasing of $h(n)$ considering the FFT and IFFT,

$$X_p = \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi}{N} np}, \quad p \in \{0, 1, 2, \dots, N - 1\}$$

$$x_n = \frac{1}{N} \sum_{p=0}^{N-1} X_p e^{j \frac{2\pi}{N} np}, \quad p \in \{0, 1, 2, \dots, N-1\}$$

Question 4.2 Design a band-pass digital filter using the Kaiser window using the following specifications:

$$f_s = 20\text{Hz}, f_{sa} = 3\text{Hz}, f_{pa} = 4\text{Hz}, f_{pb\pi} = 6\text{Hz}, f_{sb} = 8\text{Hz}$$

$$A_{pass} = 0.1\text{dB}, A_{stop} = 80\text{dB}$$

Solution

$$\delta_{pass} = \frac{10^{\frac{0.1}{20}} - 1}{10^{\frac{0.1}{20}} + 1} = 0.0058$$

$$\delta_{stop} = 10^{\frac{-80}{20}} = 0.0001$$

Therefore $\min(\delta_{pass}, \delta_{stop}) = \delta_{stop} = 0.0001A = -20\delta = A_{stop} = 80$

$$\alpha = 0.1102(A - 8.7) = 0.1102(80 - 8.7) = 7.857$$

$$D = \frac{A - 7.95}{14.36} = 5.017$$

The filter width and cut-off frequency are:

$$\Delta f = f_{stop} - f_{pass} = 1\text{Hz}$$

$$f_c = \frac{1}{2}(f_{stop} + f_{pass}) = 4.5\text{Hz}$$

$$\omega_c = \frac{2\pi f_c}{f_s} = 0.45\pi$$

The filter length is:

$$N = 1 + \frac{Df_s}{\Delta f} = 101.35 \Rightarrow N = 103, M = \frac{1}{2}(N - 1) = 51$$

The windowed impulse response for $n = 0, 1, 2, 3, \dots$

$$h(n) = w(n)d(n - m) = \frac{I_0(7.857\sqrt{n(102-n)/51})}{I_0(7.857)} \frac{\sin(0.45\pi(n-51))}{\pi(n-51)}$$

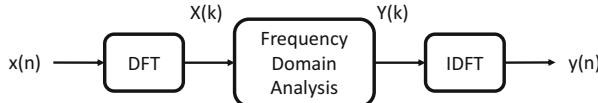


Fig. 4.2 The figure depicts the steps involved in periodic discrete-time sequence spectrum analysis using DFT

4.1.2 Bilinear Transform

One of the methods used to transform an analog filter into a digital filter (and vice versa) is Bilinear Transform. The equation below is called a bilinear transform where z corresponds to a linear transform in both denominator and numerator. In this equation the s indicates Laplace transform i.e. the system is in continuous time, and z denotes discrete time. T is the sampling period. Here the s plane is mapped on to the unit circle in the z -plane. The left half of the s plane lies in the unit circle and right side of the s plane lies outside the unit circle.

$$s = \frac{2}{T} \left(\frac{z - 1}{z + 1} \right) = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (4.12)$$

Substituting $s = j\Omega$ where Ω is the frequency in continuous time and $z = e^{j\omega}$ where ω is a discrete time frequency, the mappings between them are shown in Eqs. 4.13–4.15 (Fig. 4.2).

$$j\Omega = \frac{2}{T} \left(\frac{e^{j\omega} - 1}{e^{j\omega} + 1} \right) \quad (4.13)$$

$$\Omega = \frac{2}{T} \tan \left(\frac{\omega}{2} \right) \quad (4.14)$$

$$\omega = 2 \tan^{-1} \left(\frac{\Omega}{2} \right) \quad (4.15)$$

4.2 Windowing for Filtering

Now it is the time to discuss designing and selecting a window to ‘frame’ a signal that is to be analysed. This operation requires making a trade-off between time and frequency. Most windows are such that they only take non-negative values in both time and frequency domains. If they take negative values, these are much smaller than the positive values. In addition, they pick at the origin in both time and

frequency domains. For this type of window, it is possible to define an equivalent time width N_e and equivalent bandwidth β_e , as follows:

$$N_e = \frac{\sum_{k=-(M-1)}^{M-1} \omega(k)}{\omega(0)} \quad \text{and} \quad \beta_e = \frac{\int_{-\pi}^{+\pi} W(\omega) d\omega}{W(0)} \quad (4.16)$$

From direct and inverse DTFTs, we obtain

$$W(0) = \sum_{k=-\infty}^{+\infty} \omega(k) = \sum_{k=-(M-1)}^{M-1} \omega(k) \quad \text{and} \quad \omega(0) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} W(\omega) d\omega \quad (4.17)$$

This, from previous two equations, 4.16 and 4.17 we can conclude that the time and bandwidth product equals to unity in Eq. 4.18 and a window cannot be both time-limited and band-limited. The more slowly window decays to zero in one domain, the more consecrated it is in the other domain.

$$N_e \beta_e = 1 \quad (4.18)$$

Now, what are these window functions? The window functions are sometime domain signals which are in general concentrated on short time duration signals. These functions normally perform low frequency filtering operation. Some commonly used such functions are rectangular, Hamming, Hanning, Kaiser, Blackman etc. The operation of “windowing” means multiplying a signal by a window function. Therefore windowing signals means multiplying a signal by a window which is zero everywhere except for the region of interest. The assumption is that the signals are infinite duration but we can discard all of the resulting zeros and concentrate on just the windowed portion of the signal. In this analysis, signals are analyzed taking a small portion of the signal at a time. In the windowing process the signals are divided into a set of sections which can be termed as window or window of data or a frame or window frame. Each of these frames or windows has an equal number of data points or samples. We presume that signal is constant within the window or has a constant frequency or at most the signal is slowly varying over the time-span of the analysis. The four aspects one needs to consider in windowing are:

- **Type:** This is the name and type of the window function. Some window types are noted below in the text.
- **Size:** This is the length of the window. Denoted here as k .
- **Starting:** This defines where the window starts relative to the signal. For example if it starts at the beginning of the signal than starting window is 0.
- **Shifting:** This is a step size or shift of the window denoted by m .

Speech is a time-varying signal. Therefore, to analyze the speech signal we take a small portion of the signal at a time, and so on. There are different types of window function that facilitate windowing a signal. There are a trade-offs processing a

signal following window method, for example resolution, spectral leakage (results in discontinuities) and processing loss (loss of information). The parameters that are used to process the signal through analysis help us to obtain more information in the different sections of the signal.

4.3 Allpass Filters

The allpass filter has a spectral magnitude of unity. The spectrum of the output is the same as the spectrum of the input. The allpass filter is a complex pole-zero filter and it generates complex valued signals. The allpass filter is a mirror image of numerator and denominator i.e.

$$b[n] = a[N - n] \Leftrightarrow B(z) = z^{-N} A(z^{-1}) \Rightarrow |H(e^{j\omega})| \equiv 1 \quad \forall \omega$$

An example of allpass filter is the unit delay operator $Z = e^{j\omega}$. The phase of the unit delay operator ω is $j\omega$. The ratio between any complex number and its complex conjugate $\frac{x+jy}{x-jy}$ is of unit magnitude as $x + jy = \rho e^{j\omega}$ and $x - jy = \rho e^{-j\omega}$. This ratio equals $|e^{j2\omega}|$. Thus a minimum phase filter $H(\omega)$ can give an all pass filter $P(z)$, if $P(z) = \frac{H(\omega)}{\bar{H}(\omega)}$ such that $H(\omega) = 1 - \frac{z}{z_r}$ and $\bar{H}(\omega) = 1 - \frac{1}{zz_r}$. The all pass filter is not causal, so it is multiplied by another all-pass operator.

$$P(z) = \frac{zH\left(\frac{1}{z}\right)}{H(z)} = \frac{z - \frac{1}{z_r}}{1 - \frac{z}{z_r}} \quad (4.19)$$

The denominator has a pole at z_r and $z_r = \frac{e^{j\omega_0}}{\rho}$. The pole is outside the unit circle and the zero is inside the unit circle. Thus in an all-pass filter $P(z) = \frac{zH\left(\frac{1}{z}\right)}{H(z)} = 1$ where $H(\omega)$ is a minimum phase filter. The frequency selective filters are realized as parallel combination of two all passes filters:

$$H(z) = \frac{1}{2}[A_1(z) + A_2(z)] \quad (4.20)$$

Some digital filters are designed using the prototypes of some analog filters such as Butterworth, Chebyshev, and elliptic via the bilinear transformation and these can be characterized by using all pass sums. The digital filters designed using the prototype of all pass sums, exhibit low complexity structures, and they have linear phase, low implementation complexity and delay. Example of a first order all pass filter and a second order pass filter are shown in Figs. 4.3 and 4.4 and are written in Eqs. 4.21 and 4.22. First order all pass filter is:

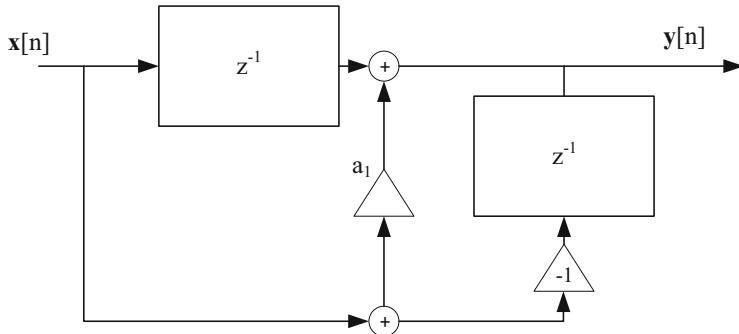


Fig. 4.3 A basic structure of one state allpass filter

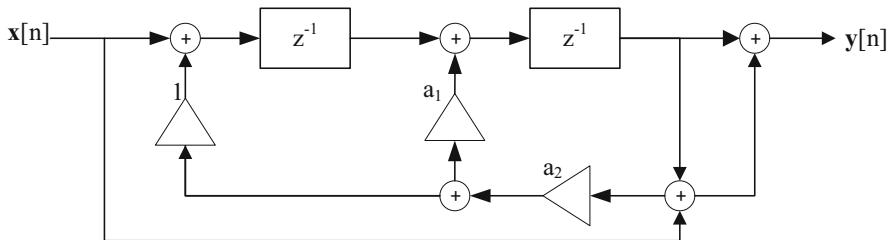


Fig. 4.4 A two stage allpass filter

$$H(z) = \frac{a_1 + z^{-1}}{1 + a_1 z^{-1}} \quad (4.21)$$

A second order all pass filter is:

$$H(z) = \frac{a_2 + a_1 z^{-1} + z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (4.22)$$

Due to low complexity structures with low round off noise behaviour, an all pass filter is applied in many applications such as phase equalization, fractional delay design, multivariate filtering, filter banks, notch filtering, recursive phase splitters etc.

4.4 Lattice Filters

Lattice Filters are a type of adaptive filter. The structure of this filter can be either *FIR* or *IIR*. One common application of this type of filter is in speech processing. Suppose G is allpass filter, then lattice filter in shown in Figs. 4.5, 4.6, and 4.7. The transfer function of the filter is written in Eqs. 4.23, 4.24, 4.25, 4.26, and 4.27.

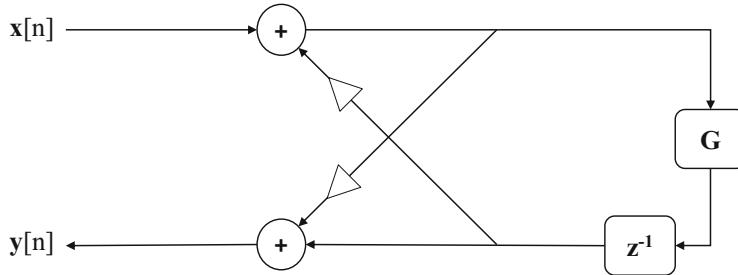


Fig. 4.5 A basic structure of allpass filter

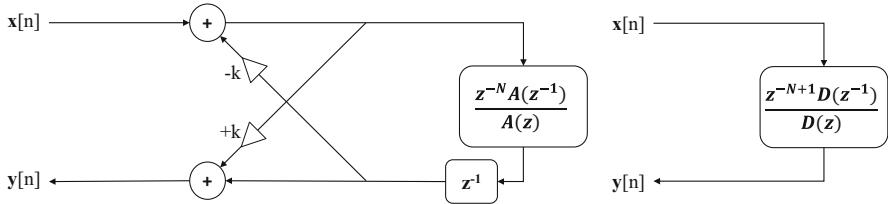


Fig. 4.6 Transformation of a lattice filter and its transfer function from $A(z)$ to $D(z)$ and $D(z)$ to $A(z)$

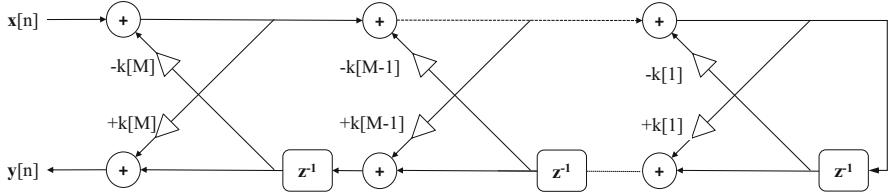


Fig. 4.7 M stage allpass lattice filter

$$G(z) = \frac{z^{-N} A(z^{-1})}{A(z)} \quad (4.23)$$

$$V(z) = X(z) - kz^{-1} G(z) V(z) \quad (4.24)$$

$$\frac{Y(z)}{X(z)} = \frac{kA(z) + z^{-N-1}A(z-1)}{A(z) + kz^{-N-1}A(z-1)} \equiv \frac{z^{-(N+1)}D(z^{-1})}{D(z)} \quad (4.25)$$

Obtaining $\{d[n]\}$ from $\{a[n]\}$:

$$d[n] = \begin{cases} 1 & \text{for } n = 0 \\ a[n] + ka[N + 1 - n] & \text{for } 1 \leq n \leq N \end{cases} \quad (4.26)$$

Obtaining $\{a[n]\}$ from $\{d[n]\}$:

$$k = d[N + 1] \quad \text{and} \quad a[n] = \frac{d[n] - kdN + 1 - n}{1 - k^2} \quad (4.27)$$

If $G(z)$ is stable then $\frac{Y(z)}{X(z)}$ is stable if and only if $|k| < 1$

An m stage allpass lattice filter is M stage of $\frac{z^{-m}A(z^{-1})}{A(z)}$ as shown in Fig. 4.7 where

$$A_M(z) + A(z) \text{ for } m = M : [-1, +1]$$

$$\text{and } k[m] = a_m[m]$$

$$a_{m-1}[n] = \frac{a_m[n] - k[m]a_m[m-n]}{1 - k^2[m]} \quad \text{for } 0 \leq n \leq m-1 \quad (4.28)$$

Equivalently:

$$A_{m-1}(z) = \frac{A_z(z) - k[m]z^{-m}A_m(z^{-1})}{1 - k^2[m]}$$

$A(z)$ is stable if and only if (iff) $|k[m]| < 1 \quad \forall m$ such that $m \in$

Now in the Fig. 4.8 is a lattice filter where $H_m(z) = \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A_m(z)}$ where $V_m(z)$ is the z -transform of $v_m[n]$ and where $U_m(z)$ is the z -transform of $u_m[n]$.

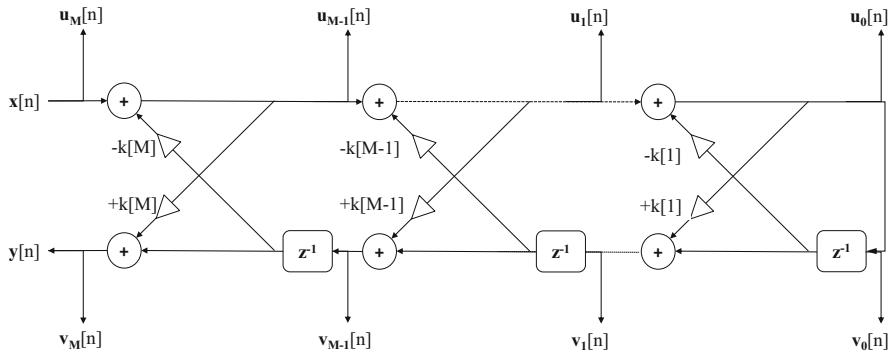


Fig. 4.8 An M stage lattice filter

$$\frac{U_{m-1}(z)}{U_m(z)} = \frac{1}{1 + k[m]z^{-1}H_{m-1}(z)} = \frac{A_{m-1}(z)}{A_{m-1}(z) + k[m]z^{-m}A_{m-1}(z^{-1})} = \frac{A_{m-1}(z)}{A_m(z)} \quad (4.29)$$

and

$$\frac{U_m(z)}{X(z)} = \frac{A_m(z)}{A(z)},$$

$$\frac{V_m(z)}{X(z)} = \frac{U_m(z)}{X(z)} \times \frac{V_m(z)}{U_m(z)} = \frac{z^{-m}A_m(z^{-1})}{A(z)} \quad (4.30)$$

Example:

$$A(z) = A_3(z) = 1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}$$

$$k[3] = 0.2 = a_2 = \frac{[1, 0.2, 0.23] - 0.2[0.2, -0.23, 0.2]}{1 - 0.2^2} = [1, 0.256, -0.281]$$

$$k[2] = -0.281 = a_1 = \frac{[1, 0.256] + 0.281[-0.281, 0.256]}{1 - 0.281^2} = [1, 0.357]$$

$$k[1] = 0.357$$

$$a_0 = 1$$

Also:

$$\frac{V_0(z)}{X(z)} = \frac{1}{1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}}$$

$$\frac{V_1(z)}{X(z)} = \frac{0.357 + z^{-1}}{1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}}$$

$$\frac{V_2(z)}{X(z)} = \frac{-0.281 + 0.256z^{-1} + z^{-2}}{1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}}$$

$$\frac{V_3(z)}{X(z)} = \frac{0.2 - 0.23z^{-1} + 0.2z^{-2} + z^{-3}}{1 + 0.2z^{-1} - 0.23z^{-2} + 0.2z^{-3}}$$

One type of filter is all-zero lattice filter which is FIR lattice filter. The other one is all pole lattice filter which is all pole filter of a type of IIR filter. Then again a combination of poles and zeros lattice typed IIR filter is commonly known as lattice ladder filter. Below we introduce this type of the filter in brief.

4.5 All-Zero Lattice Filter

This filter has a simple form depicted in the following equations. For $m = 1, 2, \dots, M - 1$

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1) \quad (4.31)$$

$$g_m(n) = K_m f_{m-1}(n) + g_{m-1}(n-1) \quad (4.32)$$

Following the similar structure, the transfer function of All-pole Lattice filter can be derived as written in Eq. 4.33:

$$H(z) = \frac{1}{1 + \sum_{k=1}^N a_N(k)z^{-k}} \quad (4.33)$$

4.6 Lattice Ladder Filters

This is an IIR filter type where both poles and zeros are lattice typed structured written in Eq. 4.34.

$$H(z) = \frac{\sum_{k=0}^M b_M(k)z^{-k}}{1 + \sum_{k=1}^N a_N(k)z^{-k}} = \frac{B_M(z)}{A_N(z)} \quad (4.34)$$

An application of this filter is detailed in Part III of this book.

4.7 Comb Filter

A basic comb filter written in Eq. 4.35 is a simple IIR comb filter. It has a one parameter α . This can be FIR or IIR filter with a feedback coefficient of α .

$$y(n) - \alpha y(n - N) = x(n) \quad (4.35)$$

The transfer function of comb filter shown Eq. 4.35 is in Eq. 4.36:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \alpha z^{-N}} \quad (4.36)$$

where N is time delay. Similarly a *FIR* comb filter is given in the filling Eq. 4.37:

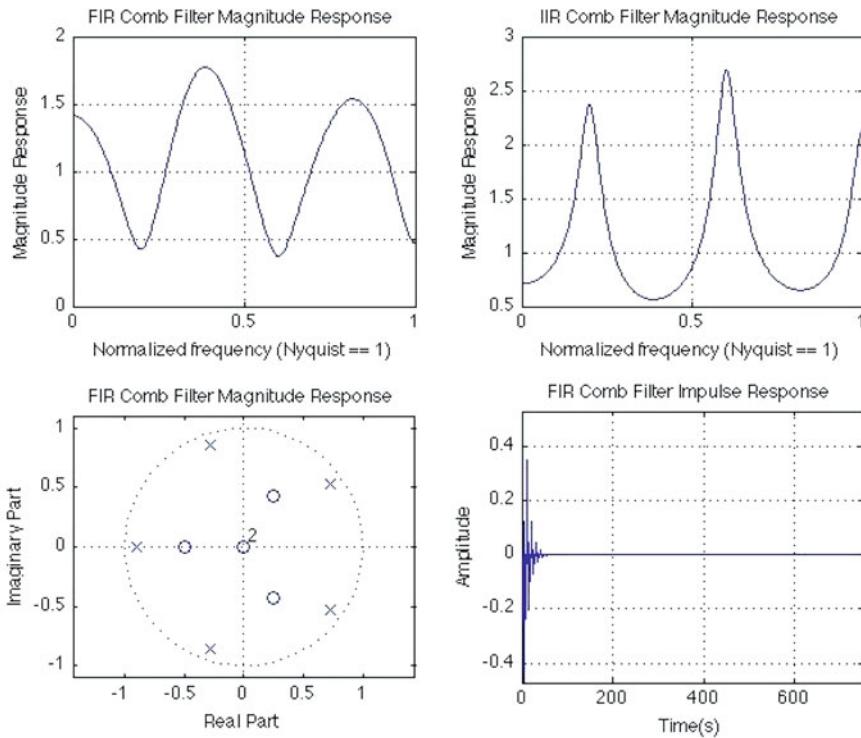


Fig. 4.9 Comb filter magnitude and frequency response

$$y(n) = x(n) + \beta x(n - N) \quad (4.37)$$

whereas the transfer function is given by Eq. 4.38.

$$H(z) = \frac{Y(z)}{X(z)} = 1 + \beta z^{-N} \quad (4.38)$$

The following comb filter is analyzed based on the zeros $\{1, 0, 0, 0.125\}$ and poles $\{1, 0, 0, 0, 0, 0.5905\}$ (Fig. 4.9).

4.8 Notch Filter

The notch filter is a type of the band stop filter. Elimination of specific frequency by digital *FIR* filter generates an overly broad stop-band and suppresses other frequencies. In such situation the notch filter with a narrow stop-band at ω_0 can be applied for specific frequency elimination. The bandwidth B which is approximately

3 dB. In Eq. 4.39, the constant ϑ is selected for the desired frequency that is distant from the frequency that is to be eliminated.

$$y(n) - 2a \cos(\omega_0)y(n-1) + a^2y(n-2) = -\vartheta(x(n) - 2\cos(\omega_0)x(n-1) + x(n-2)) \quad (4.39)$$

$$B \approx \frac{1-a}{\pi} \quad (\text{Normalized frequency in Hz}) \quad (4.40)$$

Question 4.3 Construct a digital notch filter to eliminate a 50 Hz frequency component. The stop band should have 3 dB width 50 ± 5 Hz and the sampling frequency is 500 Hz.

Solution Normalized notch frequency: $f_0 = \frac{50}{500} = 0.1$ Normalized stop-band width: $B = \frac{10}{500} = 0.02$

Coefficient $a = 1 - B\pi = 1 - 0.02\pi = 0.937$, and $\cos(\omega_0) = \cos(2\pi f_0) = 0.8090$

The notch filter is now:

$$y(n) - 2*0.937*0.809*y(n-1) + 0.875y(n-2) = x(n) - 1.618x(n-1) + x(n-2)$$

$$y(n) - 1.516y(n-1) + 0.875y(n-2) = x(n) - 1.618x(n-1) + x(n-2)$$

In Fig. 4.10, we see the pole-zero plot of a notch filter in a and its notch in the frequency response of the IIR notch filter in b at 50 Hz.

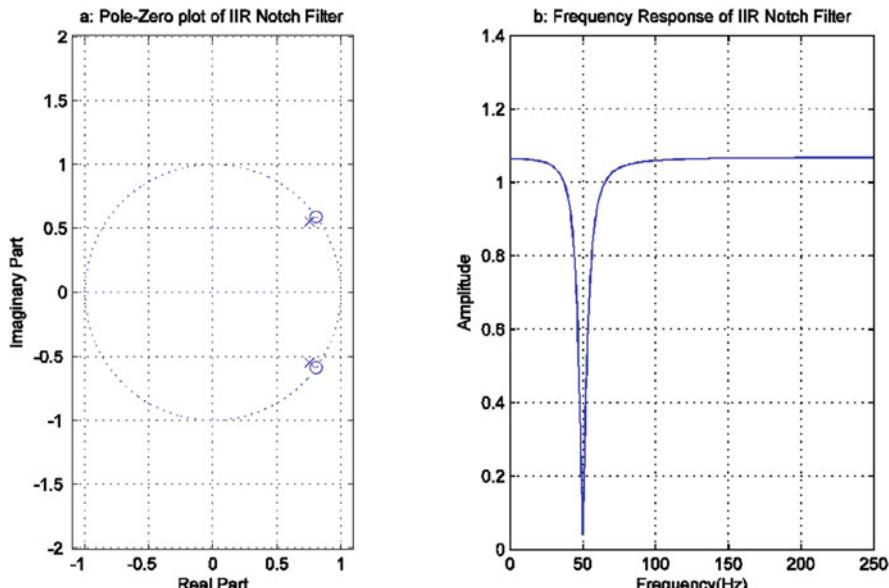


Fig. 4.10 Notch filter and its frequency response.

4.9 Background Information

Filters are some mathematical manipulations of mathematical formulations which get input and manipulate it to generate the output. The filters are structured to fit specific needs and their architecture can be of different types following different techniques. The filtering concept is applied in the time domain and in the frequency domain. In the past, the industry focus was on analog filters. Our text focuses on digital filters. The digital filters are characterized sometimes by impulse response, difference equation and different types of transformations. The filter design can be also of different types such as direct form I, direct form II, cascaded form, parallel form. The types are also different such as low-pass, high-pass, band-pass, band-stop, which can again be categorized by finite impulse response (FIR) or infinite impulse response (IIR) (Proakis 1996). The digital filter can be defied in state space form which sometimes is called adaptive filter. The simple low-pass filter such as first-order low-pass filter is commonly used in antialiasing. On the other hand, a common use of the simple high-pass is used in pre-emphasizing speech signal to compensate for radiation losses at the lips (Orfanidis 2010; Smith 2008).

4.10 Exercises

Question 4.4 Using the Hamming window, design a filter with following specifications: sampling rate 22 kHz cutoff frequency 5.7 kHz, and filter order of 315. Calculate and plot the magnitude response of the filter.

Question 4.5 Design a digital *FIR* band-pass filter using Blackman window for coefficients computing the frequency response and using the following specifications:

Passband cutoff frequency:	7 – 12 kHz
Stopband ripple:	0.001
Passband ripple:	0.001
Transition width:	3 kHz
Sampling frequency:	44.1 kHz

Question 4.6 Design a digital *IIR* low-pass 8'th order elliptic filter with the following specifications:

Passband cutoff frequency:	300 Hz
Stopband ripple:	0.001
Passband ripple:	0.5 dB
Stopband attenuation:	50 dB
Transition width:	3 kHz
Sampling frequency:	6 kHz

References

- [Proakis1996] J. G. Proakis, D. G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, 3rd Edition, Prentice Hall, 1996
- [Orfanidis2010] Orfanidis, S.J., Introduction to Signal Processing, Pearson Education Inc, 2010
- [Smith2008] Smith J. O., Introduction to Digital Filters: with Audio Applications, 2nd Edition, W3K Publishing, 2008

Chapter 5

Estimation and Detection



Overview

In this chapter we will explain how adaptive filters are used in stochastic signal processing and estimation as well as detection. Real world signals are commonly random. The typical *FIR* or *IIR* filter cannot handle the random characteristics. Adaptive filters are better suited for stochastic and random signal processing. This type of signal processing is non-linear and is more complex than what is encountered with linear filters. One of the most common adaptive filtering approaches is the least square estimation. This approach minimizes an error computed as a difference between the predicted signal and measured signal. The applications of adaptive filters are very wide and the procedures of applying adaptive filters can be different from one area to another. Some adaptive filters are used for estimation, while others for predictions. Other common adaptive filters are the Wiener filter, Kalman filter, extended Kalman filter, particle filter, etc.

5.1 Introduction

With adaptive filters, the input signal goes as an input into the filter where it is used as reference when compared with the generated output signal. The difference between these two signals is the error signal. This error signal provides the performance measure.

In a dual perspective shown in Fig. 5.1, the input signal goes through the filter where it is processed based on the difference between the current output and a reference signal. The reference signal can be a set of desired parameters or properties of the output. The goal is to generate the output which closely approximates the desired signal and where the error between the output and the desired signal is the smallest.

Mathematical techniques to obtain such output are briefly described next.

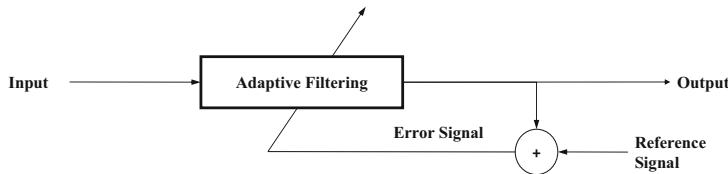


Fig. 5.1 Standard adaptive signal processing approach

Adaptive filtering refers to the process by which the generated output is tuned to be more similar to the input signal which shows randomness as time changes. As new information comes in, the adaptive filters update its parameters or the coefficients. In order to apply adaptive filtering, the output signal constraints are well defined. This means that priori information has to be given to the system. Function of thus prior information, the filter can more or less accurately represent the input signal. In this chapter, we will show how the desired signal is represented by the filter, and how the filter updates the coefficients in order to generate the best close approximate of the input signal.

In estimation, no priori information about the value of some parameter θ is given. Hypothesis testing is a related problem, a hypothesis specifies a value for the parameter θ . An estimator is a prescription for using the data to find a value for the parameter θ . The estimator is defined by a random variable. A particular estimate is found by substituting the realization of the data; it is called the θ estimate.

5.2 Hypothesis Testing

Uncertainty is one of the major issues that is handled in statistical signal processing. Signal detection theory analyzes observations, measurements which are affected by environmental factors. For making decisions, probabilities lay the foundations for handling uncertainty using decision theory. The uncertainty is handled with hypotheses.

5.2.1 Bayesian Hypothesis Testing

In Bayes statistical inference, the unknown events are considered random and described by a priori distributions. With a binary hypothesis the randomly generated output of a source has two different alternatives, such as the hypothesis H_0 and H_1 .

The prior probabilities are:

- $p_0 = P[H_0]$; this means

$P[\text{success of, for instance, event } A \text{ is 0 and it has not occurred}]$

- $p_1 = (1 - p_0) = P[H_1]$; this means

$P[\text{success of, for instance, event } A \text{ is 1 and it has occurred}]$

If a conditional probability density function of some random variable X with respect to some hypothesis H , aka likelihood of X , is denoted by $p_{x|H}$, then the related conditional densities of an observation x of X with regard to hypotheses H_0 and H_1 are $p_{x|H}(x|H_0)$ and $p_{x|H}(x|H_1)$. The Maximum a Posteriori (MAP) probability method minimizes the probability of error and decides about the more likely of the two hypotheses. In this case, for an observation x of a random variable X ,

If $P[X = x; H_0] > P[X = x; H_1]$, then decide to prefer H_0 ,

If $P[X = x; H_0] < P[X = x; H_1]$, then decide to prefer H_1 ,

If $P[X = x; H_0] = P[X = x; H_1]$, then decide arbitrarily to prefer either H_0 or H_1 .

The MAP criteria for probability density functions with Bayesian hypotheses is:

$$\begin{aligned} L(x) &\cong \frac{p_{x|H}(x|H_1)}{p_{x|H}(x|H_0)} < \left(\frac{p_0}{p_1} \right) \Rightarrow \text{decide to prefer } H_0 \\ L(x) &\cong \frac{p_{x|H}(x|H_1)}{p_{x|H}(x|H_0)} > \left(\frac{p_0}{p_1} \right) \Rightarrow \text{decide to prefer } H_1 \end{aligned} \quad (5.1)$$

In a non-Bayesian hypotheses case, hypotheses H_0 and H_1 are not random but rather deterministic events. They are not assigned probabilities. Thus, only the original formulation applies, with the semantic:

$$P[\text{event } A \text{ occurred when } H_0 \text{ is true}] = P[A; H_0] \quad (5.2)$$

An application of binary non-Bayesian hypothesis testing is introduced in Part III, where it is applied to strong noise handling using matched filtering.

5.2.2 MAP Hypothesis Testing

For the observation $X = x$, and a priory probabilities p_0 and p_1 for hypotheses, the posterior probability is:

$$\begin{aligned} P(H_0|x) &= P(H_0 = \text{true}|X = x), \\ P(H_1|x) &= P(H_1 = \text{true}|X = x) \end{aligned} \quad (5.3)$$

The minimum error probability test in Eq. 5.3 is a **Maximum A Posteriori** (MAP) hypothesis testing criterion, where $L(x)$ is the likelihood ratio:

$$L(x) \cong \frac{p_{x|H}(H_1|x)}{p_{x|H}(H_0|x)} > \left(\frac{p_1}{p_0} \right) \Rightarrow \text{decide about } H_1 \quad (5.4)$$

5.3 Maximum Likelihood (ML) Hypothesis Testing

In **Maximum Likelihood** (ML) hypothesis testing, the prior hypothesis probabilities p_0 and p_1 are assumed equal. If

$$p_0 = p_1 = \frac{1}{2} \Rightarrow L(x) \cong \frac{p(H_1)}{p(H_0)} > \left(\frac{p_1}{p_0} \right) = 1 \Rightarrow \text{decide to prefer } H_1 \quad (5.5)$$

The Eq. 5.5 is the maximum likelihood criterion, denoted *ML* criterion, for the hypothesis testing and it is sometimes used instead of MAP as a more efficient alternative.

5.4 Standard Estimation Techniques

First, the signal is approximated with a parameterized function using some optimization technique, in a parametric signal modeling phase. The common types of signal modeling techniques are: Auto-regressive technique (*AR*), Auto-regressive and moving average (*ARMA*) technique, moving average (*MA*) technique. In addition some aspects of Bayesian reasoning are used.

These parameters are transferred to the adaptive filters and the filter uses some optimization technique such as:

- i. Minimum Variance Unbiased Technique (MVU)
- ii. Best Linear Unbiased Estimator (BLUE)
- iii. Maximum Likelihood Estimator (MLE)
- iv. Least Squares Estimator (LSE)

The Bayesian estimators are:

- i. Minimum or Least Mean Square error Estimators (LMSE) or (MMSE)
- ii. Linear Minimum Mean Square error Estimators (LMMSE)

Below we describe a brief theoretical aspect of these methods and their applications.

Suppose $x_1, x_2, x_3, \dots, x_n$ is a time series of some random variables $X_1, X_2, X_3, \dots, X_n$ modeling the evolution of a random variable in time by replication at fix

intervals and the parameter is θ . The variables $X_1, X_2, X_3, \dots, X_n$ are identically distributed and $p(x|\theta)$ is the joint probability density at $x = (x_1, x_2, x_3, \dots, x_n)$. Thus the mathematical formulation of this joint probability density function for a random signal is:

$$p(x, \theta) = \prod_{i=1}^n p(x_i, \theta) \quad (5.6)$$

The log likelihood is the natural logarithm of the likelihood:

$$l(\theta, x) = \log L(\theta, x) \quad (5.7)$$

The estimation of an unknown parameter θ is investigated from a set of observations \mathbf{x} . The observation might be noisy or corrupted and the desired input can be mixed with some interference.

If x_n is an observation at certain time n and $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^T$ which is a vector of length N , the problem is to estimate θ which is $\hat{\theta}$ such that

$$\hat{\theta} = g(\mathbf{x}) = g(x_0, x_1, \dots, x_{N-1}); \text{ where } g(\mathbf{x}) \text{ is an estimator function.} \quad (5.8)$$

If $\hat{\theta}$ is closest to the ground truth θ , then the estimation is optimal.

Below we are introducing some commonly used estimation methods.

The $X_1, X_2, X_3, \dots, X_n$ are independent variables following a probability density function $f(x, \theta)$. This *pdf* can be discrete when we talk about Poisson distribution and the *pdf* can be continuous when we talk about normal or exponential distribution.

Now, a statistic is in the first place an arbitrary function $T(X_1, X_2, X_3, \dots, X_n)$ of the data $X_1, X_2, X_3, \dots, X_n$ such that $T(X)$ for $X = X_1, X_2, X_3, \dots, X_n$ depend on $X_1, X_2, X_3, \dots, X_n$ but does not depend on θ . Such a statistic $T(X)$ can be used as an estimator of the parameter θ .

If $X_1, X_2, X_3, \dots, X_n$ are independent and identically distributed (i.i.d.), the statistic $T(X)$ is an unbiased estimator of θ if for all values of θ , $E_\theta(T(X)) = E(\hat{\theta}) = \theta$. The estimator is unbiased if its probability distribution is always centered at the true value of the parameter.

If $\hat{\theta}$ is not an unbiased estimator of θ , then the bias estimator of θ is the difference: $E(\hat{\theta}) - \theta$ of $\hat{\theta}$, that is

$$bias(\hat{\theta}) = E_{\hat{\theta}}(T(X)) = E(\hat{\theta}) - \theta \quad \forall \text{ values of } \theta$$

5.4.1 Minimum Variance Unbiased (MVU) Estimator

The minimum variance unbiased estimator is an unbiased estimator which has lower variance than all the other unbiased estimators. A common estimation measurement criteria is the Mean Squared Error (MSE) criteria which is the expectation of the minimum square difference between $\hat{\theta}$ and θ with respect to $\hat{\theta}$ such that:

$$\hat{\theta} = E[(\hat{\theta} - E(\hat{\theta})) + (E(\hat{\theta}) - \theta)]^2 = \text{var}(\hat{\theta}) + [E(\hat{\theta}) - \theta]^2 \quad (5.9)$$

The expression $[E(\hat{\theta}) - \theta]^2$ is a function of θ . The variance of the estimator $\text{var}(\hat{\theta})$ is the only component which is a function of the data. When this is minimized, $[E(\hat{\theta}) - \theta] = 0$ and this minimizes $\text{var}(\hat{\theta})$. This is called MVU estimation. The MVU estimator is commonly minimum and unbiased only in certain range [a,b]. Namely:

$$E(\hat{\theta}) = \theta \text{ for } a < \theta < b \quad (5.10)$$

The estimator $\hat{\theta}$ must be of minimum variance:

$$\hat{\theta}_{mvu} = \hat{\theta} \underset{\theta}{\operatorname{argmin}} \left\{ \text{var}(\hat{\theta}) \right\} = \hat{\theta} \underset{\theta}{\operatorname{argmin}} \left\{ E(\hat{\theta} - E(\hat{\theta}))^2 \right\} \quad (5.11)$$

A Gamma distribution and a Poisson process type of distribution are example of MVU estimations.

Example 5.1 Suppose $X_1, X_2, X_3, \dots, X_n$ are independent and identically distributed (i.i.d.) random variables. What is an unbiased estimator of θ ?

The unbiased estimator of θ is $\bar{\theta}$:

$$\hat{\theta} = \frac{2}{N} \sum_{i=1}^N x_i$$

Example 5.2 Suppose $X_1, X_2, X_3, \dots, X_n$ are independent Poisson (λ) random variables for $i = 1, 2, 3, \dots, n$ and an unbiased estimator of λ . Also $S = \sum_{i=1}^n X_i$ is a sufficient static for λ .

The unbiased estimator is given by any $E(\sum_{i=1}^n X_i), \dots, E(X_n | \sum_{i=1}^n X_i)$.

$$E(X_1 \sum_{i=1}^n X_i) + E(X_2 \sum_{i=1}^n X_i) + \dots + E(X_n \sum_{i=1}^n X_i) = E[\sum_{i=1}^n X_i | \sum_{j=1}^n X_j] = \sum_{i=1}^n X_i \quad (5.12)$$

$$nE(\sum_{i=1}^n X_i) = \sum_{i=1}^n X_i U = E(\sum_{i=1}^n X_i) = \underline{X}$$

\underline{X} is a MVUE of λ . As its variance is equal to the Cramer-Rao Lower Bound for λ .

5.4.2 Best Linear Unbiased Estimator (BLUE)

The **Best Linear Unbiased Estimator** (BLUE) is based on the ordinary least squares estimator. It has the lowest variance of the estimate with respect to the unbiased and linear estimator. If an estimator is linear, minimum and unbiased, then the estimator is BLUE. Since the estimator is linear, $\hat{\theta}$ can be written as:

$$\hat{\theta} = A \quad (5.13)$$

The unbiased estimation based on Eq. 5.13 is written

$$E(\hat{\theta}) = AE(x) = \theta \quad (5.14)$$

and it satisfies

$$E(x) = H\theta \quad (5.15)$$

such that

$$AH = I \quad (5.16)$$

BLUE is derived by finding \mathbf{A} which minimizes the variance $C_{\hat{\theta}} = ACA^T$ and $C = E[(x - E(x))(x - E(x))^T]$ is co-variance of x . The BLUE is

$$\hat{\theta} = Ax = (X^T C^{-1} H)^{-1} H^T C^{-1} x \quad (5.17)$$

The minimum variance is

$$C_{\hat{\theta}} = (X^T C^{-1} H)^{-1} \quad (5.18)$$

Example 5.3 An example of such estimator is Gauss-Markov process estimator.

$$x = H\theta + w \quad (5.19)$$

Where, H is know, w is noise with co-variance C , and the BLUE is:

$$\hat{\theta} = (X^T C^{-1} H)^{-1} H^T C^{-1} x \quad (5.20)$$

The minimum variance in Eq. 5.20 is

$$C_{\hat{\theta}} = (X^T C^{-1} H)^{-1} \quad (5.21)$$

5.4.3 Maximum Likelihood Estimator (MLE)

The MLE estimates one or more unknown parameters. Suppose $Y_0, Y_1, Y_2, \dots, Y_{N-1}$ is a series of random variables describing the possible readings of a process \mathbf{Y} and $y_0, y_1, y_2, y_3, \dots, y_{n-1}$ are obtained sample values. If these readings are independent and identically distributed (IID) such that the observations $y_0, y_1, y_2, y_3, \dots, y_{n-1}$ have joint density function denoted as:

$$p(y_0, y_1, y_2, y_3, \dots, y_{n-1})$$

The likelihood of $y_0 = Y_0, y_1 = Y_1, y_2 = Y_2, \dots, y_{n-1} = Y_{n-1}$ given θ is the function

$$like(\theta) = p(y_0, y_1, y_2, y_3, \dots, y_{n-1} | \theta)$$

and can be processed as a function of θ . The maximum likelihood estimate of θ is the value of this function that maximizes $like(\theta)$. It is the value for parameters that makes the observed data the most probable. Now we can write based on the independence assumption:

$$p(\mathbf{Y} | \theta) = \prod_{k=1}^N p(y_k | \theta)$$

In maximum likelihood, the parameters are assumed to be fixed but unknown. The MLE seeks the solution that best explains the dataset \mathbf{Y} . The likelihood conveys the same meaning when it is used for discrete probability distributions. The maximum likelihood estimator finds the parameter that maximizes the probability density of the data or the measurement.

MLE maximizes the likelihood function $L(\theta)$ with respect to unknown parameter θ . In MLE, the logarithm of the likelihood function is actually used, because the logarithm is a monotonic increasing function and transforms the product in sum. The MLE estimator is the maximizing the $l(\theta)$ (see Eq. 5.22) with respect to θ .

$$l(\theta) = \log L(\theta) = \prod_{k=1}^N p(y_k | \theta) = \sum_{k=1}^N \log(p(y_k | \theta)) \quad (5.22)$$

Example 5.4 Suppose a series of readings from a discrete random variable X with domain $[0, 1, 2, 3]$ and the probability distribution $P(X) = [\frac{2\theta}{3}, \frac{\theta}{3}, \frac{2(1-\theta)}{3}, \frac{(1-\theta)}{3}]$. At 10 measurements the observations are $(3, 0, 2, 1, 3, 2, 1, 0, 2, 1)$. Compute the maximum likelihood estimate of θ .

Solution: Since the observations are: $(3, 0, 2, 1, 3, 2, 1, 0, 2, 1)$:

$$L(\theta) = p(X = 3)p(X = 0)p(X = 2)p(X = 1)p(X = 3)$$

$$p(X=2)p(X=1)p(X=0)p(X=2)p(X=1)$$

Substituting from the probability distributions one obtains:

$$L(\theta) = \prod_{i=1}^n p(X=x_i|\theta) = \left(\frac{2\theta}{3}\right)^2 \left(\frac{\theta}{3}\right)^3 \left(\frac{2(1-\theta)}{3}\right)^3 \left(\frac{(1-\theta)}{3}\right)^2$$

Maximizing the likelihood function $L(\theta)$ is equivalent to maximizing the \log of $L(\theta)$ denoted as $l(\theta)$. Thus the likelihood function is:

$$\begin{aligned} l(\theta) = \log L(\theta) &= \sum_{i=1}^n \log p(\theta) = \\ &2\left(\log \frac{2}{3} + \log(\theta)\right) + 3\left(\frac{1}{3} + \log(\theta)\right) + \\ &3\left(\frac{2}{3} + \log(1-\theta)\right) + 2\left(\frac{1}{3} + \log(1-\theta)\right) = \\ &C + 5\log(\theta) + 5\log(1-\theta) \end{aligned}$$

We will set the derivative of $l(\theta)$ with respect to θ to be 0/zero:

$$\frac{\partial l(\theta)}{\partial \theta} = \frac{5}{\theta} - \frac{5}{1-\theta} = 0$$

Thus, the MLE of θ is $\hat{\theta} = 0.5$

5.4.4 Least Squares Estimator (LSE)

The least squares estimation is about estimating the parameters by minimizing the squared magnitudes of the difference between the observed measurements and the signal. This method searches the a and b which determine the best fit of $y = ax + b$ given a set of observations (x_n, y_n) for all x_n where data has been measured or, with prediction, where it could be measured.

An approach is to find the best fit of a finite combinations of specified functions f_1, f_2, \dots, f_n . There the main problem is to find the coefficients a_1, a_2, \dots, a_n such that Eq. 5.23 best approximates or predicts of the data.

$$y = a_1 f_1(x) + a_2 f_2(x) + \dots + a_n f_n(x) \quad (5.23)$$

Applications of this method are expanded in Part III of the book.

Example 5.5 Assume the data points $(-6, -1)$, $(-2, 2)$, $(1, 1)$, and $(7, 6)$. Find a least squares solution of $X[b \ a] = Y$ for $X = [[1 \ -6] [1 \ -2] [1 \ 1] [1 \ 7]]$ and $b = [-1 \ 2 \ 1 \ 6]$.

The columns a_1 and a_2 of matrix X are orthogonal. The orthogonal projection of Y onto $\text{Col } X$ is given by:

$$\hat{Y} = \frac{Y \cdot a_1}{a_1 \cdot a_1} a_1 + \frac{Y \cdot a_2}{a_2 \cdot a_2} a_2 = \frac{8}{4} a_1 + \frac{45}{90} a_2 =$$

$$[2 \ 2 \ 2 \ 2] + \left[-3 - 1 \frac{1}{2} \frac{7}{2} \right] = \left[-11 \frac{5}{2} \frac{11}{2} \right]$$

Now that \hat{Y} is known, we can solve $X[\hat{b} \ \hat{a}] = \hat{Y}$. The weights of X generate \hat{Y} .

$$[\hat{b} \ \hat{a}] = \left[\frac{8}{4} \ \frac{45}{90} \right] = \left[2 \ \frac{1}{2} \right]$$

Minimum or least mean square error estimator (LMSE) or (MMSE) given $X_1 = x_1, \dots, X_L = x_L$, is the conditional expectation of \mathbf{Y} . In Eq. 5.24, $\hat{y} = E[Y]$.

$$E[(Y - \hat{y})^2] = E[(Y - E[Y])^2] = \sigma_Y^2 \quad (5.24)$$

The **Minimal Mean Square Error Estimator** on $\hat{y}(x)$ as written in Eq. 5.25 where the MMSE for the given value of X is the conditional variance, i.e. the variance $\sigma_{(Y|X)}^2$ of the conditional density of $f_{(Y|X)}(X)$.

$$\hat{y}(x) = \int_{-\infty}^{+\infty} y f_{Y|X}(X = x) dy = E[Y|X = x] \quad (5.25)$$

Example 5.6 Example of an MMSE Estimator for a signal in additive noise: Suppose the random variable X is a noisy measurement of the angular position Y of an antenna, so $X = Y + W$, where W denotes the additive noise. Assume the noise is independent of the angular position, i.e., Y and W are independent random variables, with Y uniformly distributed in the interval $[-1, 1]$ and W uniformly distributed in the interval $[-2, 2]$.

Solution: Given that $X = x$, we would like to determine the MMSE estimate $\hat{y}(x)$, the resulting mean square error, and the overall mean square error averaged over all possible values x that the random variable X can take.

$\hat{y}(x)$ is the resulting mean square error, and the overall mean square error averaged over all possible values x that the random variable X can take. Since $\hat{y}(x)$ is the conditional expectation of Y given $X = x$, we need to determine $f_{Y|X}(y|x)$. For this, first determine the joint density of Y and W , and from this the required conditional density. For the independent Y and W :

$$f_{Y,W}(y|w) = f_Y(y) f_W(w) = \begin{cases} \frac{1}{8} & -2 \leq w \leq 2 \& -1 \leq y \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (5.26)$$

Conditioned on $Y = y$, X is the same as $y + W$, uniformly distributed over the interval $[y - 2, y + 2]$, hence:

$$f_{(X,Y)}(x, y) = f_{X|Y}(y) f_Y(y) = \left(\frac{1}{4}\right) \left(\frac{1}{2}\right) = \frac{1}{8} \quad (5.27)$$

The function is defined on range of values given by $-1 \leq y \leq 1$ $y - 2 \leq x \leq y + 2$, and 0 otherwise. The joint pdf is therefore uniform over a parallelogram.

Given $X = x$, the conditional pdf $f_{(Y|X)}$ is uniform on the corresponding vertical section of the parallelogram:

$$f_{(Y,X)}(y, x) = \begin{cases} \frac{1}{3+x} & \text{for } -3 \leq x \leq -1 \text{ and } -1 \leq y \leq x + 2 \\ \frac{1}{2} & \text{for } -1 \leq x \leq 1 \quad \text{and } -1 \leq y \leq 1 \\ \frac{1}{3-x} & \text{for } 1 \leq x \leq 3 \quad \text{and } -2 \leq y \leq 1 \end{cases} \quad (5.28)$$

The MMSE estimates $\hat{y}(x)$ is the conditional mean of Y given $X = x$, and the conditional mean is the midpoint of the corresponding vertical section of the parallelogram.

$$\hat{y}(x) = E[Y|X = x] = \begin{cases} \frac{1}{2} + \frac{1}{2}x & \text{for } -3 \leq x \leq -1 \\ 0 & \text{for } -1 \leq x \leq 1 \\ -\frac{1}{2} + \frac{1}{2}x & \text{for } 1 \leq x \leq 3 \end{cases} \quad (5.29)$$

The MMSE associated with the estimate is the variance of the uniform distribution Eq. 5.28, specifically

$$E[\{Y - \hat{y}(x)\}^2|X = x] = \begin{cases} \frac{(3+x)^2}{12} & \text{for } -3 \leq x \leq -1 \\ \frac{1}{3} & \text{for } -1 \leq x \leq 1 \\ \frac{(3-x)^2}{12} & \text{for } 1 \leq x \leq 3 \end{cases} \quad (5.30)$$

Equation 5.30 specifies the mean square error that results for any specific value x of the measurement of X . Since the measurement is a random variable, the marginal pdf of X is:

$$f_X(x) = \frac{f_{X,Y}(x, y)}{f_{Y,X}(y, x)} = \begin{cases} \frac{(3+x)}{8} & \text{for } -3 \leq x \leq -1 \\ \frac{1}{4} & \text{for } -1 \leq x \leq 1 \\ \frac{(3-x)}{8} & \text{for } 1 \leq x \leq 3 \end{cases} \quad (5.31)$$

Now by convolution, $f_X = f_Y * f_W$ as Y , and W are statically independent, the,

$$E_X[E_{Y|X}\{Y - \hat{y}(x)\}^2|X = x] = \int_{-\infty}^{+\infty} E[\{Y - \hat{y}(x)\}^2|X = x] f_X(x) dx =$$

$$\int_{-3}^{-1} \frac{(3+x)^2}{12} \frac{(3+x)}{8} dx + \int_{-1}^{+1} \frac{1}{3} \frac{1}{4} dx + \int_1^3 \frac{(3-x)^2}{12} \frac{(3-x)}{8} dx \quad (5.32)$$

The mean square error is the variance of Y i.e. σ_Y^2 in Eq. 5.33 below and the mean square error of the estimated Y by its mean is 0. Thus the MMSE of the estimator is in Eq. 5.33.

$$\sigma_Y^2 = \frac{[1 - (-1)]^2}{12} = \frac{1}{3} \quad (5.33)$$

5.4.5 Linear Minimum Mean Square Error Estimator (LMMSE)

The conditional expectation $E(Y|X)$ in MMSE is difficult to determine, because the conditional density $f_{(Y|X)}(y|x)$ is not easily determined. Thus an easier and efficient approach is to let the estimator to be a fixed linear function of the measured random variables, and to choose the linear relationship so as to minimize the mean square error. The resulting estimator is called the [Linear Minimal Mean Square Error Estimator](#) (LMMSE). In this manner, the random variable Y in terms of another random variable X , restricting the estimator to be of the form in Eq. 5.34, where a and b are to be determined so as to minimize the mean square error as in Eq. 5.35.

$$\hat{Y}_l = \hat{y}_l(X) = aX + b \quad (5.34)$$

In Eq. 5.34, the expectation is taken over the joint density of Y and X , the linear estimator is optimum when averaged over all possible combinations of Y and X that may occur. Once the a and b in Eq. 5.35 is chosen, the estimate of Y for a particular x , is just $\hat{y}_l(x) = ax + b$ for particular values of a and b . Thus in LMMSE, a linear optimal estimator generates an estimator for any particular x that this estimator generates an estimate that is the optimal, where in MMSE, an optimal MMSE estimator for each x is $E[Y|X = x]$, that minimized the mean square error conditioned on $X = x$.

$$E_{Y,X}[(Y - \hat{Y}_l)^2] = E_{Y,X}[(Y - (aX + b))^2] \quad (5.35)$$

The distinction between MMSE and LMMSE is that, in MMSE, the optimal estimator is obtained by joining together all the individual optimal estimates, whereas in the LMMSE estimates are obtained by simply evaluating the optimal linear estimator.

Example 5.7 Example of LMMSE Estimator for signals in additive noise: Following the same example provided for the MMSE estimator, we can now design an LMMSE estimator. The random variable X denotes a noisy measurement of the

angular position Y of an antenna, so $X = Y + W$, where W denotes the additive noise. We assume the noise is independent of the angular position, i.e., Y and W are independent random variables, with Y of the angular position, i.e. Y and W are independent random variables, with Y uniformly distributed in the interval $[-1, 1]$ and W uniformly distributed in the interval $[-2, 2]$.

For the LMMSE, the estimate of Y in terms of X , we need to determine the respective means and variances, as well as the covariance, of these random variables. Now we arrive in Eqs. 5.36, and 5.37,

$$\mu_Y = 0, \mu_W = 0, \mu_X = 0, \sigma_Y^2 = \frac{1}{3}, \sigma_W^2 = \frac{4}{3} \quad (5.36)$$

$$\sigma_X^2 = \sigma_Y^2 + \sigma_W^2 = \frac{5}{3}, \sigma_{YX} = \sigma_Y^2 = \frac{1}{3}, \rho_{YX} = \frac{1}{\sqrt{5}} \quad (5.37)$$

The LMMSE estimator is accordingly formulated by Eq. 5.38:

$$\hat{Y}_l = \frac{1}{5}X \quad (5.38)$$

and the associated MMSE is written by Eq. 5.39:

$$\sigma_Y^2(1 - \rho^2) = \frac{4}{15} \quad (5.39)$$

Below we describe a brief theoretical aspect of these estimators and their applications.

In general, the adaptive signals come with uncertainty, their properties change with time. Statistical interferences, estimations, approximation, optimizations, and probabilistic transforms are applied to process adaptive signals. The adaptive filter is a type of self-adjusting digital filter. The adaptive filters adjust the filter coefficients using adaptive algorithms in order to adapt the input signal. The adaptive filters are time variant, and the parameters of adaptive filters are continuously changing with time in order to meet performance requirements (Barkat 2005).

An adaptive filter plays an important and effective role, when the specifications are unknown, or cannot be managed using time-invariant filters. An adaptive filter is often a non-linear filter, in which characteristics depend on the signal, and linear properties such as homogeneity, additivity are not satisfied. But the adaptive filters can be used as linear filters. The input signal, reference signal, signal model, filter designed are some essential components in adaptive filtering. Based on the input, and reference signal, the signal model, and designed filter, the parameters are updated through an algorithm. Thus the solution approach to the adaptive filtering problem is mostly data driven (Poor 1994).

Applications of adaptive signal processing are very wide. Some basic applications of adaptive filtering are echo cancellation, equalization, system identification, noise cancellation, beam forming. The adaptive filter can be structured as a

transversal filter or an IIR filter. The IIR filter structures can be cascade, lattice or parallel. A suitable algorithm is very important to adjust the coefficients, or the parameters of the adaptive filtering, and to minimize the error, or the optimization criteria fulfilled. The algorithm can be based on the searching or minimizations of certain functions, or the objective, or error function. The choice of algorithm determines the several crucial aspects of the adaptive process, such as existence of certain aspects or conditions of optimal solutions, computational complexities (Togneri 2005). In adaptive filtering, a common optimization approach is the error criterion, which may be based on Mean Square Error (MSE), Least Squares (LS), Weighted Least Squared (WLS), Minimum Mean Square Error (MMSE). These were briefly introduced in this chapter.

5.5 Exercises

Question 5.1 Suppose $X_1, X_2, X_3, \dots, X_n$ are i.i.d. random variables with density function $p(\sigma) = \frac{1}{2\sigma} \exp\left(-\frac{|X|}{\sigma}\right)$. Find the maximum likelihood estimate of σ . How can you use *R* or *matlab* or *python* to find the MLE.

Question 5.2 Suppose $X_1, X_2, X_3, \dots, X_n$ are i.i.d. random variable with Poisson distribution with μ such that $\mu > 0$. The likelihood function is $L(\mu; x) = \prod_{i=1}^n \frac{(e^{-\mu} \mu^{x_i})}{x_i!}$. How can you use *R* or *matlab* or *python* to find the MLE.

Question 5.3 Suppose $X_1, X_2, X_3, \dots, X_n$ are i.i.d. random variable with Gamma distribution with unknown parameters α and β . Determine ML estimator for α and β . How can you use *R* or *matlab* or *python* to find the MLE.

Question 5.4 Suppose $X = (X_1, X_2, X_3, \dots, X_n)^T$ be a random variable with Gamma distribution as $\text{Gamma}(\lambda, \alpha)$ such that:

$$f(x : \lambda, \alpha) = \frac{\lambda^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-\lambda y}$$

Identify the complete statics of $(\lambda, \alpha)^T$

References

- [Togneri2005] Togneri, R. Estimation Theory for Engineers, Lecture Notes, August 30, 2005.
- [Barkat2005] Barkat, M., Signal Detection and Estimation, 2nd Edition, Artech House, 2005
- [Poor1994] Poor, H. V., An Introduction to Signal Detection and Estimation, 2nd Edition, Springer-Verlag, 1994

Chapter 6

Adaptive Signal Processing



Overview

Signal modeling refers to representing the signal via the parameters of some model. For example, signals are modeled as sums of sinusoids, estimating the parameters of these sinusoids, and storing the parameters instead of the original samples. The application of signal modeling is signal compression, prediction, reconstruction, feature extraction, information extraction, and recognition and identification. Some standard signal modelings are ARMA, MA, AR. There is also low rank signal modeling which represents the signal samples through low rank matrices. These are used in computer vision, data mining, and bioinformatics. In low rank modeling, the matrix completion and dimensionality reduction is important. In image compression, the JPEG coding can be mentioned as a well known application of signal modeling. Adaptive filters, and their applications, parametric, and non-parametric signal are discussed in this chapter.

6.1 Introduction

The basic random signal processing concepts from the perspective of modelling are shown in Fig. 6.1. The signal x is an impulse or white noise. x is filtered by H , which can be for example an ARMA family such as Auto-Regressive (AR), Moving Average (MA) and Auto-Regressive Moving Average (ARMA), and generate the filtered signal \hat{x} . H can be deterministic autocorrelation of the filter's response.

In signal model identification, signal is given as samples at times $0, 1, 2, \dots, N-1$, and the filter is specified via parameters. Model parameters are those, that match the best with the estimated signal values.

The spectral estimation problem estimates the total distributed power over frequency from a finite record of a stationary data sequence. Two common such estimations are the spectral analysis by parametric estimation and the spectral

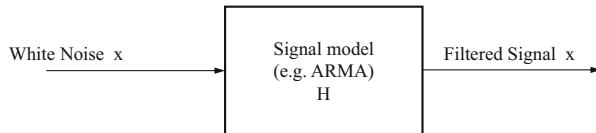


Fig. 6.1 Stochastic signal processing

analysis by nonparametric estimation. In non-parametric approximation, the signal is applied to a band pass filter with a narrow bandwidth. The desired frequency goes through and the filter output power divided by the filter bandwidth is used as a measure of the spectral content of the input to the filter.

In parametric approach, the model has to fit the data for an accurate estimation. In non-parametric approach, the spectral estimation is directly applied on the data.

6.2 Parametric Signal Modeling

In parametric signal modeling, the signal is represented in such a way that the complicated process turns into a simpler process especially into a smaller number of parameters which eventually represent the signal. A linear prediction based on least squares is a common approach for parametric signal modelling.

6.2.1 Parametric Estimation

In the parametric estimation, a process is represented optimally by a finite number of parameters. These parameters are extracted by a number of techniques. Here we mainly focus on the techniques used for the adaptive filtering. The estimator is applied to the data. When the parameters are estimated using prior distributions, then it is known as Bayesian Estimation. The central focus of such estimation is to find the posterior distribution based on the prior distribution.

We see the adaptive filtering and how these are applied for the real world applications in the next sections.

6.3 Wiener Filtering

The Wiener filtering is an effective method when the noise is known and if it is stationary.

$$x \equiv (\dots, x_{-1}, x_0, x_1, \dots) = (x_k, k \in \mathbb{Z})$$

$$x_k \in R^{d_x} \text{ (vector valued signal).}$$

Its Fourier (two-sided), and Z transform, are given by Eqs. 6.1 and 6.2 below.

$$X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x_k e^{-j\omega k}, \quad \omega \in [-\pi, \pi] \quad (6.1)$$

$$X(z) = \sum_{k=-\infty}^{\infty} x_k z^{-k}, \quad z \in D_x \quad (6.2)$$

We are given two processes:

- s_k , the signal to be estimated
- y_k , the observed process

They are jointly wide-sense stationary, with known covariance functions: $R_s(k)$, $R_n(k)$, $R_{sn}(k)$.

A particular case is that of a signal corrupted by additive noise, depicted in Eq. 6.3:

$$y_k = s_k + n_k \quad (6.3)$$

Estimate s_k as a function of y . Specifically: Find the linear MMSE estimate of s_k based on y_k . There are three versions of this problem introduced in Chap. 3:

- Causal Filter: $\hat{s}_k = \sum_{m=-\infty}^k h_{k-m} y_m$
- Non-causal Filter: $\hat{s}_k = \sum_{m=-\infty}^{k+l} h_{k-m} y_m$
- FIR Filter: $\hat{s}_k = \sum_{m=k-M}^k h_{k-m} y_m$

Assuming that *FIR* filter is of the length $N + 1$:

$$\hat{s}_k = \sum_{m=-\infty}^k h_{k-m} y_m = \sum_{i=0}^k h_i y_{k-i} \quad (6.4)$$

We need to find the coefficients (h_i) that minimize the MSE Eq. 6.5.

$$E[(s_k - \hat{s}_k)^2] \rightarrow \min \quad (6.5)$$

To find (h_i) , we can differentiate the error. More conveniently, start with orthogonality principle in Eq. 6.6.

$$E[(s_k - \hat{s}_k)] = 0; j = 0, 1, 2, \dots, N \quad (6.6)$$

This gives Eqs. 6.7 and 6.8 given below

$$\sum_{i=0}^N E[y_{k-i} y_{k-j}] = E(s_k y_{k-j}) \quad (6.7)$$

$$\sum_{i=0}^N h_i R_y(i-j) = R_{sy}(j) \quad (6.8)$$

In matrix form it is denoted by Eq. 6.9

$$\begin{pmatrix} R_y(0) & R_y(1) & \cdots & R_y(N) \\ R_y(1) & R_y(0) & \cdots & R_y(N-1) \\ R_y(2) & R_y(1) & \cdots & R_y(N-2) \\ \vdots & & & \\ R_y(N) & R_y(N-1) & \cdots & R_y(0) \end{pmatrix} [h_0 \cdots h_N] = [R_{sy}(0) \cdots R_{sy}(N)] \quad (6.9)$$

$$R_y h = r_{sy} \Rightarrow h = R_y^{-1} r_{sy} \quad (6.10)$$

The last two equations that were presented are known as the Yule-Walker equations.

We note that $R_y \geq 0$ is positive semi-definite, and non-singular matrix except for degenerate cases. Further more, it is a Toeplitz matrix for which there exist efficient algorithms (Levinson-Durbin and others) that utilize the matrix special structure to compute h . The structure of the matrix is such that it is:

- Symmetric about the diagonal,
- All elements of the diagonal are equal, and
- Matrix is (always) invertible

The MMSE may now be easily computed by the equation provided next:

$$E[(s_k - \hat{s}_k)^2] = E[(s_k - \hat{s}_k)(-s_k)] \quad (6.11)$$

$$= R_s(0) - E(s_k \hat{s}_k) \quad (6.12)$$

$$= R_s(0) - h^T r_{sy} \quad (6.13)$$

Assume, \hat{s}_k may depend on the entire observation signal, $y_k, k \in Z$. The FIR approach fails when the transversal filter has infinitely many coefficients.

The required filter form is in Eq. 6.14, presented below:

$$\hat{s}_k = \sum_{i=-\infty}^{\infty} h_i y_{k-i} \quad (6.14)$$

Orthogonality implies: $(\hat{s}_k - s_k)$ is perpendicular to y_{k-j} such that $j \in Z$. Therefore, Eqs. 6.14 and 6.15 hold:

$$E(s_k y_{k-j}) = E(\hat{s}_k y_{k-j}) = \sum_{i=-\infty}^{\infty} h_i R(j-i) \quad -\infty < j < +\infty \quad (6.15)$$

6.4 Kalman Filter

The mean squared filter estimator of $x(k+1), \hat{x}(k+1)|(k+1)$, in predictor-corrector format is formulated by Eq. 6.16.

$$\hat{x}(k+1)|(k+1) = \hat{x}(k+1)|(k) + K(k+1)\tilde{z}(k+1|k) \quad (6.16)$$

For $k = 0, 1, \dots$ where $\hat{x}(0) = m_x(0)$ and $\tilde{z}(k)$ is the innovations sequence. Kalman gain matrix $K(k+1)$ is a matrix of order $n \times m$, and is specified by the set of relations formulated in Eqs. 6.17–6.19.

$$K(k+1) = P(k)H'(k+1)[H(k+1)P(k)H'(k+1) + R(k+1)]' \quad (6.17)$$

$$P(k) = \phi(k+1, k)P(k)\phi'(k+1, k) + \tau(k+1, k)Q(k)\tau'(k+1, k) \quad (6.18)$$

and

$$P(k+1) = [I - K(k+1)H(k+1)]P(k+1|k) \quad (6.19)$$

For $k = 0, 1, \dots$, where I is the $n \times n$ identity matrix, and $P(0) = P_x(0)$.

The Kalman Filter (KF) involves feedback and contains within its structure a model of the plant. The feedback nature of the KF manifests itself in two different ways: in the calculation of $\hat{x}(k+1)|(k+1)$ and also in the calculation of the matrix of gains, $\hat{x}(k+1)$ and also in the calculation of the matrix of gains, $K(k+1)$. Predictor equations $\hat{x}(k+1|k)$ and $P(k+1|k)$, use information only from the state equation, whereas the corrector equations, compute $K(k+1)$, $\hat{x}(k+1)|(k+1)$, and $P(k+1)|(k+1)$, use information only from the measurement equation. Once the gain is computed, Eq. 6.19 represents a time-varying recursive digital filter. The result is in Eq. 6.20.

$$\begin{aligned} \hat{x}(k+1) &= [I - K(k+1)H(k+1)]\phi(k+1, k)\hat{x}(k) + \\ &K(k+1)z(k+1) + [I - K(k+1)H(k+1)]\Psi(k+1, k)u(k) \end{aligned} \quad (6.20)$$

This is a state equation for state vector \hat{x} whose time-varying-plant matrix is $[I - K(k+1)H(k+1)]\phi(k+1, k)$. Equation 6.7 is time-varying even if the basic state-variable model is time-invariant and stationary, because the gain matrix $K(k+1)$ is time-varying.

1) is still time-varying in this case. It is possible, however, for $K(k+1)$ to reach a limiting value (i.e. steady-state value, \underline{K} , in which case Eq. 6.7 reduces to a recursive constant coefficient filter.

6.4.1 Smoothing

There are three types of smoothers, the most useful one for digital signal processing is the fixed-interval smoother. The fixed-interval smoother is $\hat{x}(N)$, $k = 0, 1, 2, \dots, N-1$, where N is a fixed positive integer. The situation is here as follows: with an experiment completed, we have measurements available over the fixed interval $1 \leq k \leq N$. For each point within this interval we wish to obtain the optimal estimate of the state vector $x(k)$, which is based on all the available measurement data $z(j)$, $j = 1, 2, 3, \dots, N$. The fixed-interval smoothing is very useful in signal processing situations, where the processing is done after all the data are collected. It can not be carried out on-line during an experiment like filtering can. Because all the available data are used, we cannot hope to do better by other forms of smoothing than by fixed-interval smoothing. A mean squared fixed-interval smoothed estimate of $x(k), \hat{x}(N)$ is Eq. 6.21.

$$\hat{x}(N) = \hat{x}(k+1) + P(k-1)r(k|N) \quad k = N-1, N-2, \dots, 1 \quad (6.21)$$

and $n \times 1$ vector r satisfies the backward-recursive Eq. 6.22.

$$r(N) = [\phi'_p(j+1, k)r(N) + H'(j)P(j-1)H'(j) + R(j)]^{-1} + \tilde{z}(j|j-1) \quad (6.22)$$

where $\phi_p(k+1, k) = \phi_p(k+1, k)[I - K(k)H(k)]$ and $k = N-1, N-2, \dots, 2, 1$ and $n \times 1$ vector r satisfies the backward-recursive Eq. 6.22. The smoothing error-covariance $P(k|N)$, is Eq. 6.23.

$$P(N) = P(k-1) - P(k-1)S(k)P(k-1) \quad (6.23)$$

Where $k = N-1, N-2, \dots, 2, 1$, and $n \times n$ matrix $S(j|N)$, which is the covariance matrix of $r(j|N)$, satisfies the backward-recursive Eq. 6.24 below:

$$S(N) = \phi'_p(j+1)S(j+1|N)\phi_p(j+1, j) + H'(j)[H(j)P(j-1)H'(j) + R(j)]^{-1}H(j) \quad (6.24)$$

where $j = N, N-1, \dots, 2, 1$, and $n \times n$ matrix $S(N) = 0$.

The fixed interval smoothing involves a forward pass over the data, using a Kalman-Filter (KF), and then a backward pass over the innovations, using Eq. 6.22. The smoothing error-covariance matrix $P(N)$, can be precomputed, but it is not used during the computation of $\hat{x}[k|N]$. It is quite different than the active use of the filtering error-covariance matrix in the KF .

Equation 6.25 is a deconvolution where the system's input $\mu(j)$ is the white noise and $h(j)$ is the system's impulse response.

$$z(k) = \sum_{i=1}^k \mu(i)h(k-i) + \vartheta(k), \quad k = 1, 2, \dots, N \quad (6.25)$$

In the deconvolution process, the effects of $h(j)$ and $\vartheta(j)$ are removed in order to estimate $\mu(j)$ only. For this, Eq. 6.25 is first transformed into state variables. The single channel state-variable model is in Eqs. 6.26 and 6.26, where $x(0) = 0$, $\mu(0) = 0$, $h(0) = 0$, and $h(l) = h'\phi^{l-1} \gamma (l = 1, 2, \dots, N)$.

$$x(k+1) = \phi x(k) + \gamma \mu(k) \quad (6.26)$$

$$az(k) = h'x(k) + \vartheta(k) \quad (6.27)$$

A two-pass fixed interval smoother for Eq. 6.28 is:

$$\mu(k) \text{ is } \hat{\mu}(N) = q(k)\gamma' r(k+1|N) \text{ where } k = N-1, N-2, \dots, 3, 2, 1 \quad (6.28)$$

The smoothing error-variance is shown in Eq. 6.29.

$$(N) = q(k) - q(k)\gamma' S(N)\gamma q(k) \quad (6.29)$$

where $r(N)$ and $S(N)$ are used to obtain in Eq. 6.30.

$$E\{\mu^2(k)\} = q(k) \quad (6.30)$$

6.5 Particle Filter

The particle filtering represents a posterior *pdf* by a set of randomly chosen weighted samples such as by a **Sequential Importance Sampling (SIS)**, **Monte Carlo (MC)** process. The particle filter can solve the estimation problem online. Some significant applications of particle filtering are its use in non-linear and non-Gaussian process. Some common techniques to develop particle filtering are **condensation algorithms**, **bootstrap filtering**, **interacting particle approximations**, **particle filtering survival of the fittest**, and the **sequential Monte Carlo** approach. We will briefly introduce the *SIS* and *Monte Carlo* approach for particle filtering. The importance sampling (*IS*) is a basic particle filter building tool.

The particle filtering can represent any arbitrary distribution such as multimodal support, keep tract of several hypotheses simultaneously, approximate representation of complex model rather than exact representation of simplified model.

Particle filtering is a general Monte Carlo (sampling) method for performing inference in state-space models where the state of the system evolves in time and information about the state is obtained from the noisy observation at each time step. In general, the discrete-time state space model, the state of a system evolves as formulated in Eq. 6.31.

$$x_k = f_k(x_{k-1}, v_{k-1}) \quad (6.31)$$

In Eq. 6.31, x_k is a vector representing the state of the system at time k , $v_{(k-1)}$ is the noise vector, f_k is the possible non-linear and time-dependent function describing the evolution of the state vector. The state vector x_k is assumed to be latent variable or unobservable. Information about x_k is about obtained only through noise measurement of it, z_k , which is governed by Eq. 6.32 presented next.

$$z_k = h_k(x_k, n_k) \quad (6.32)$$

In this equation, h_k is a possibly non-linear and time-dependent function describing the measurement process and n_k is the measurement noise vector. The filtering problem involves the estimation of the state vector at time k , given all the measurements up to and including time k , which we denote by $z_{(1:k)}$. In Bayesian setting, this problem can be formalized as computation of the distribution $p(x_k|z_{1:k})$, which can be done recursively in two steps. In the prediction step, $p(x_k|z_{1:k-1})$ is computed from the filtering distribution $p(x_{k-1}|z_{1:k})$ at time $k-1$ formulated in Eq. 6.33 below.

$$p(x_k|z_{1:k}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \quad (6.33)$$

where $p(x_{k-1}|z_{1:k-1})$ is assumed to be known due to recursion and $p(x_k|x_{k-1})$ in Eq. 6.33.

The distribution $p(x_k|z_{1:k-1})$ can be thought of as a prior over x_k before receiving the most recent measurement z_k . In the update step, the prior is updated with the new measurement z_k using Bayes' rule to obtain the posterior over x_k shown in Eq. 6.34.

$$p(x_k|z_{1:k})\alpha p(z_k|x_k)p(x_{k-1}|z_{1:k-1}) \quad (6.34)$$

Equation 6.34 is approximated using the Monte Carlo sampling approximation method, because the computations in the update state written in Eqs. 6.27 and 6.28 cannot be approximated by Bayesian approximation approach. As the number of samples becomes very large, the characterization becomes an equivalent representation of the true *pdf*.

In particle filtering, the state of the system is observed as it changes over time. If we have noisy observations, we want to know the best possible estimate of the hidden variables. Some applications of particle filtering is tracking of aircraft positions from radar, estimating communications signals from noisy measurements, acoustic events localization and detection, predicting economical data, tracking of people or cars in surveillance.

6.6 Fundamentals of Monte Carlo

Initially, consider approximating a generic probability density $\pi_n(x_{1:n})$ for some fixed n .

The Monte Carlo integration evaluates the complex integrals using probabilistic techniques. In this approach, one assumes we are trying to estimate a complicated integral of a function f over some domain D as $F = \int f(\vec{x}) d\vec{x}$ and it assumes that there exists some PDF ρ defined over D . The Monte Carlo integration over the domain D is written by Eqs. 6.35 and 6.36.

$$F = \int f(\vec{x}) d\vec{x} = \int \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x}) d\vec{x} \quad (6.35)$$

However,

$$\int \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x}) d\vec{x} = E\left[\frac{f(\vec{x})}{p(\vec{x})}\right], \quad x \approx p \quad (6.36)$$

This is true for any PDF ρ defined over D . If we have i.i.d random samples $\vec{x}_0, \vec{x}_1, \dots, \vec{x}_N$ sampled from ρ , then we can approximate $E\left[\frac{f(\vec{x})}{p(\vec{x})}\right]$ by Eq. 6.37:

$$F_N = \frac{1}{N} \sum_{i=1}^N \left[\frac{f(\vec{x}_i)}{p(\vec{x}_i)} \right] \quad (6.37)$$

Guaranteed by law of large numbers shown in Eq. 6.38:

$$N \rightarrow \infty, F_N \rightarrow E\left[\frac{f(\vec{x})}{p(\vec{x})}\right] \quad (6.38)$$

6.6.1 Importance Sampling (IS)

If $p(\vec{x})$ is 0 or very small, then $\frac{f}{p}$ can be arbitrarily large, ‘damaging’ the average design ρ such that $\frac{f}{p}$ is bounded; thus ρ is similar to f as possible and where ρ is called the important or proposal density. The effects gets more samples in important areas of f , i.e. where f is large.

The IS relies on the introduction of an importance density $q_n(x_{1:n})$ written by Eq. 6.39

$$x_n(x_{1:n}) > 0 \Rightarrow q_n(x_{1:n}) > 0 \quad (6.39)$$

The IS identities are written by Eqs. 6.40 and 6.41

$$\pi_n(x_{1:n}) = \frac{\omega_n(x_{1:n})q_n(x_{1:n})}{z_n} \quad (6.40)$$

$$Z_n = \int \omega_n(x_{1:n})q_n(x_{1:n})dx_{1:n} \quad (6.41)$$

where $\omega_n(x_{1:n})$ is the normalized weight function such that the formula in Eq. 6.42 presented below holds

$$\omega_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})} \quad (6.42)$$

Convergence of IS relies on importance density $q_n(x_{1:n})$ from which it is easy to draw samples; e.g. a multivariate Gaussian. The sequential Monte Carlo methods (*SMC*) approximate the distributions using a large number of samples or particles. As the number of particles N increases toward ∞ , this converges to the actual distribution. Sample sequentially from sequence of probability densities $\pi_n(x_{1:n})$. We sample N independent random variables $X_{1:n}^i$ from each probability distribution and estimate the distribution by Eq. 6.43

$$\hat{\pi}_n(x_{1:n}) = \frac{1}{N} \sum_{i=1}^N \delta_{x_{1:n}^i}(x_{1:n}) \quad (6.43)$$

where $\delta_{x_{1:n}^i}(x_{1:n})$ is the delta at each sample. The problems is that it is difficult to sample from complex distributions, furthermore sampling is computationally complex at least linear with n . In order to solve the problem of sampling from complex distribution it is required to use an important density and weighting to model density using Eqs. 6.44 and 6.45 provided below.

$$\hat{\pi}_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n} \quad (6.44)$$

$$\hat{\pi}_n(x_{1:n}) = \frac{\omega_n(x_{1:n})q_n l_{1Ln}}{Z_n} \quad (6.45)$$

where $w_n(x_{1:n})$ are weights, $q_n(l_{1:n})$ is importance density, and Z_n is a normalization factor.

In SMC methods, the density can be estimated by Eq. 6.46

$$\hat{\pi}_n(x_{1:n}) = \frac{1}{n} \sum_{i=1}^N W_n^i \delta_{x_{1:n}^i}(x_{1:n}) \quad (6.46)$$

where

$$W_n^i = \frac{w_n(X_{1:n}^i)}{\sum_{j=1}^N w_n(X_{1:n}^j)} \quad (6.47)$$

Note, we want to pick q so that the variance is minimized and pick something close to π . In sequential importance sampling, to select importance distribution so that Eq. 6.48 holds.

$$q_n(x_{1:n}) = q_{n-1}(x_{1:n-1})q_n(x_{1:n}|x_{1:n-1}) \quad (6.48)$$

At first time step, sample from original distribution initial probability $q_1(x_1)$. For subsequent steps, pick from the conditional probabilities $q_k(x_k|X_{1:k-1}^i)$. The variance of these estimations often increases with n , so further refinement is needed. To reduce variance, sample again from the newly created approximation distributions. Each particle is associated with a number of “offspring” samples to estimate the distribution. In sequential algorithm, at time $n = 1$:

1. Complete $X_1^i \approx q_1(x_1)$
2. Compute weights $w_1(X_1^i)$ and W_1^i
3. Resample $\{W_1^i, X_1^i\}$ to obtain N equal-weight particles $\left\{ \frac{1}{N}, \underline{X}_1^i \right\}$

At time $n \geq 2$:

- (a) Sample $X_n^i \approx q_n(x_n|\underline{X}_{n-1}^i)$ and set $X_{1:n}^i \leftarrow (\underline{X}_{1:n-1}^i, X_n^i)$
- (b) Compute weights $\alpha_n(X_{1:n}^i)$ and W_n^i
- (c) If needed resample $\left\{ W_n^i, X_{1:n}^i \right\}$ to obtain $\left\{ \frac{1}{N}, \underline{X}_{1:n}^i \right\}$

Let the distribution to be modeled by Eq. 6.49

$$\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n}) \quad (6.49)$$

We can find the importance density as formulated in Eq. 6.50

$$q_n(x_{1:n-1}) = q(x_n|y_n, x_{1:n-1}) \quad (6.50)$$

and weights as they are written in equation

$$\alpha_n(x_{1:n}) = q(y_n|x_{1:n}) \quad (6.51)$$

This allows us to create a particle filter algorithm for estimating the posterior distribution. Note that q need only depend on the previous state and current observation. The particle algorithm is as follows: At $n = 1$:

4. Sample $X_1^i \approx q_1(x_1|y_1)$
5. Compute weights $w_1(X_1^i) = \frac{\mu(X_1^i)g(y_1|X_1^i)}{q(X_1^i|y_1)}$
6. Resample $\{W_1^i, X_1^i\}$ to obtain N equal-weight particles $\left\{ \frac{1}{N}, \underline{X}_1^i \right\}$
For $n \geq 2$:
 - (a) Sample $X_n^i \approx q_n(x_n|y_n, \underline{X}_{n-1}^i)$ and set $X_{1:n}^i \leftarrow (\underline{X}_{1:n-1}^i, X_n^i)$
 - (b) Compute weights $\alpha_n(X_{n-1:n}^i) = \frac{g(y_n|X_n^i)f(X_n^i|X_{n-1}^i)}{g(X_n^i|y_n, \underline{X}_{n-1}^i)}$ and W_n^i
 - (c) If needed resample $\left\{ W_n^i, X_{1:n}^i \right\}$ to obtain N equal-weights particles $\left\{ \frac{1}{N}, \underline{X}_{1:n}^i \right\}$

At each time step, we obtain estimate distributions: where $\delta_{x_{1:n}^i}(x_{1:n})$ is the delta at each sample. The problems are it is difficult to sample from complex distributions and sampling is computationally complex at least linear with n . In order to solve the problem of sampling from complex distribution is to use an importance density and weighting to model density. The model densities are formulated in Eqs. 6.52 and 6.53.

$$\hat{p}_n(x_{1:n}) = \frac{1}{N} \sum_{i=1}^N \delta_{x_{1:n}^i}(x_{1:n}) \quad (6.52)$$

$$\hat{p}(y_n|y_{1:n-1}) = \sum_{i=1}^N W_{x_{n-1}^i} \alpha_n X_{n-1:n}^i \quad (6.53)$$

Even if optimal importance distribution is used, the model may not be efficient, and only the most recent particles are sampled at time n . The basic particle filtering algorithm is enhanced by resampling before computing weights i.e. from independence and this is auxiliary particle filtering, resample-move algorithm using Markov chain Monte Carlo (MCMC), that introduces diversity, but the same number of resampling steps, further blocking sampling resamples previous components in blocks.

6.7 Non-Parametric Signal Modeling

In this modeling, power spectral density (PSD or psd) is estimated. The Fourier transform is commonly used to estimate the psd of the signal. This type of estimation is generally estimating the autocorrelation function of the signal as the psd, and it is the common preliminary spectral analysis tool.

Periodogram, correlogram etc. are used to search the hidden periodicities of the data. To find the periodicities of the data $\{x(n)\}_0^{N-1}$, first we compute the autocorrelation sequence $r(k)$ for $k \leq (N - 1)$ and then we take the DTFT, i.e. in Eq. 6.54, the periodogram is the squared magnitude of the DTFT of the observation.

$$P(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n} \right|^2 \quad (6.54)$$

In practical scenario, the periodogram is calculated by applying the FFT i.e. depicted in the Eq. 6.55 below

$$P(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi f n}{N}} \right|^2 \quad (6.55)$$

To allow finer frequency spacing periodogram, the process is typically zero padded, i.e. the Eq. 6.56.

$$P(f_k) = \frac{1}{N'} \left| \sum_{n=0}^{N'-1} x(n) e^{-\frac{j2\pi f n}{N'}} \right|^2 \quad (6.56)$$

The performance of the periodogram does not improve for large number of samples i.e. N . A good estimator estimates better when the observed data samples increase. Theoretically as n gets larger, the estimator should get more and more data samples and it is desired that the estimated PSD tends to the true value of the PSD, i.e. if for finite number of data samples, the estimator is biased, the bias should tend to zero as $N \rightarrow \infty$ as should the variance of the estimate. The periodogram is asymptotically unbiased, as it is not a consistent estimator.

6.8 Non-Parametric Estimation

The periodogram and correlogram are some common power spectrum density estimation tools. They are good for primary evaluations but they do not work well when the data are large and they are based on the weak condition which assume the first moment and the autocovariance do not vary with time of the series.

6.8.1 Correlogram

The correlation based definition based on Eqs. 6.57 and 6.58 is the Power Spectral Density (PSD) leads to the correlogram spectral estimator. In the equation, $\hat{r}(k)$ denotes an estimate of the covariance lag $r(k)$ of the samples $\{y(1), y(2), \dots, y(N)\}$.

$$\hat{\varphi}_c(\omega) = \sum_{k=-(N-1)}^{N-1} \hat{r}(k) e^{-j\omega k} \quad (6.57)$$

$$\hat{r}(k) = \frac{1}{N-k} \sum_{t=k+1}^N y(t)y^*(t-k), \quad 0 \leq k \leq N-1 \quad (6.58)$$

6.8.2 Periodogram

The periodogram identifies the dominant characteristics such as hidden periodicities, peak frequencies, of data in the spectral domain. In syntactical sense, it can be used as the information of the periodicities of the data. The periodogram gives information about the relative strengths of the various frequencies for explaining the variation in the data. It is defined on Eq. 6.59 of samples $y(1), y(2), \dots, y(N)$. The periodogram is applied on the DFT of the data.

$$\hat{\varphi}_c(\omega) = \frac{1}{N} \left| \sum_{t=1}^N y(t) e^{-j\omega t} \right|^2 \quad (6.59)$$

The periodogram and correlogram as PSD estimators can be used to improve the PSD of the time series. Two common methods in periodogram are bias analysis and variance analysis. In the approaches, the total squared error of the power spectral estimate is the sum of the bias squared and the variance. In Eq. 6.60, a denotes any quantities to be estimated and \hat{a} is the estimate of a . The mean squared error (MSE) of the estimate is computed by Eq. 6.60.

$$\begin{aligned} MSE &\equiv E\{|\hat{a} - a|^2\} = E\{|\hat{a} - E\{\hat{a}\} + E\{\hat{a}\} - a|^2\} = \\ &= E\{|\hat{a} - E\{\hat{a}\}|^2\} - |E\{\hat{a}\} - a|^2 + 2Re[E\{\hat{a} - E\{\hat{a}\}(E\{\hat{a}\} - a)\}] = \\ &= var\{\hat{a}\} + |bias\{\hat{a}\}|^2 \end{aligned} \quad (6.60)$$

Now the bias and variance are analyzed and estimated by different window methods for computing the periodogram. One of the most common windows for this is Hamming window and this is frequently used in speech signal. Two such examples applying Black-Tukey window and Welsch window are given below.

6.9 Filter Bank Method

A filterbank consist of filters which separate a signal into frequency bands, where the sub-band transformation is the decomposition of a signal into a set of different frequency components or subband. An example of two-channel filterbank is shown in Fig. 6.2. In this Figure, the discrete time signal $x[n]$ enters the analysis bank and filtered by the filters $L(z)$ and $H(z)$, which separates the frequency content of the input signal in frequency band of equal width. The filters $L(z)$ and $H(z)$ are therefore respectively a low-pass and a high-pass filter. The output of the filters each contains half the frequency content, but an equal amount of samples as the input signal. The two outputs together contain the same frequency content as the input signal, however the amount of data is doubled. Therefore, downsampling by a factor of two, denoted $\downarrow 2$, is applied to the outputs of the filters in the analysis filter bank. The reconstruction of the original signal is possible using the synthesis filter bank. In the synthesis bank, the signals are upsampled by two, denoted by $\uparrow 2$, are passed to the filters $L^*(z)$ and $H^*(z)$. The filters are the synthesis bank based on the filters in the analysis bank. The outputs of the filters in the synthesis bank is summed, leading to the reconstructed signal $\tilde{x} \sim [n]$. The different output signals of the analysis filter bank are called subbands, the filter-bank technique is called the sub-band coding.

The filter banks can be arrangements of low pass, bandpass, and highpass filters used for the spectral decomposition and composition of signals. In such arrangements, the filter sweeps through the interval of interest, this is seen as a bank of filters. They have many useful applications such as in audio and image signal processing. Some applications in audio signal processing applying filter banks are perceptual audio coding, multi-band equalizer, machine hearing and audio

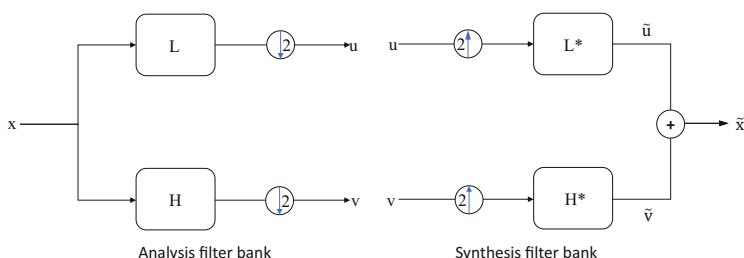


Fig. 6.2 A basic structure of two channel filter bank

component analysis. The reason of their popularity is the fact that they easily allow the extraction of spectral components of a signal while providing very efficient implementations. Since most filterbanks use various sampling rate, therefore, the filterbank is also called the multivariate signal processing. In this approach, the sample rate is changed to perform efficient signal processing operations and to construct filters with decreasing and increasing the sampling rate of the signal for the same input output rate. There are various multivariate filtering techniques such as uniform DFT filterbanks, multivariate signal processing operations. The M filters are used to estimate the spectral contents of the signal, then it is typically called as M-band, or, M-subb, or M-subbands, or M-Channel filter banks. It may retain M analysis filters with different passbands. Each of the subbband is decomposed by M-analysis filters with different passbands. Thus, each of the subband signals carries information on the input signal in a particular frequency band. They decompose the given signal into several frequency channels or bands and to reconstruct the signal from the individual channel information. Some common processing the decomposition and synthesis is down sampling by using N factor, similarly upsampling using where $M, N \equiv Z$. If $M = N$, i.e. the sampling rate conversion and the number of the channels is equal, then it is called critical sampling. In this sampling, the maximum downsampling factor for which perfect reconstruction can be achieved. Perfect reconstruction means that the output signal is a copy of the input signal with no further distortion than a time shift and amplitude scaling. the synthesized signal reconstructed the similar as the original as shown in Eq. 6.61 below:

$$x \approx \tilde{x} \quad (6.61)$$

If the following conditions are fulfilled by formulations 6.62

$$\begin{aligned} L(-z)L^*(z) + H(-z)H^*(z) &= 0 \\ L(z)L^*(z) + H(z)H^*(z) &= 2 \end{aligned} \quad (6.62)$$

This is about multivariate filterbank where the bases are constructed. The idea of such systems is to take an input system and split it into subsequences using bank of filters. Splitting can be done in two to more parts. Each part is called channel or subband.

Here data are analyzed in a finite dimensional space and power spectral density is investigated. The power distribution is investigated from a finite length data sequence. Here the signal is splitted into a number of subset using number of filters in a filter bank. Few such filter banks are first discussed.

In a perfect reconstruction, Eq. 6.63 holds.

$$\hat{x}[n] = cx[n - n_0] \quad (6.63)$$

A reconstruction is exactly done by a multiplicative constant and a delay. Otherwise three possible types of distortion such as aliasing, amplitude and phase distortion of the input and reconstructed or synthesized signal will occur.

The wavelet coding and subband coding have many similarities. Traditionally the subband coding uses filters that have little overlap to isolate different bands. The wavelet transform imposes smoothness conditions on the filters that usually represent a set of basis generated by shifting and scaling or dilation of a mother wavelet function. The wavelet can be motivated from overcoming the poor time-domain localization of short time Fourier transform.

In the analysis bank, the low pass filter and the high pass filter are downsampled by factor 2 in the decimation. When the upper half-band i.e. $\left[\frac{f_s}{4}, \frac{f_s}{2}\right]$ is decimated, it is aliased or mirrored to the lower frequencies $\left[0, \frac{f_s}{2}\right]$. f_s is the sampling frequency. The aliasing does not corrupt spectral information since the lower frequency components were filtered out using a high pass filter. In the synthesis bank, the low pass filter and the high pass filter is upsampled in the interpolation where zeros are added between the sample values in the signal by multiplying the signal by the same upsampling factor in order to keep the level unchanged. During spectral analysis, the spectrum of the signal high passed and decimated in analysis bank and in the synthesis bank, the spectrum is interpolated and high pass filtered. The higher half-band is reconstructed despite the decimation at the subband and the entire original signal can be reconstructed by summing the upper and lower half bands at the output of the synthesis bank. Two subbands are not used in practice, rather good for examples in order to understand the basic of subband analysis. The principle scales easily to n uniformly distributed subbands. Here n subbands and each decimated by factor k . Each bandwidth is equal in uniformly distributed subband coding and the decimation and interpolation factor is also the same factor k . Normally they are critically distributed where the number of bands n and the interpolation or the decimation factor k is the same i.e. $n = k$. In uniformly distributed subband coding, in the analysis bank, if the number of bands is n and the spectrum in range $\left(0, \frac{f_s}{2}\right)$ is divided into n bands, each of width $\left(\frac{f_s}{2n}\right)$. The band pass filter $H_m(f)$ in the Analysis bank selects band m which covers the frequencies $\left[\frac{mf_s}{2n}, \frac{(m+1)f_s}{2n}\right]$, $\forall m = 0, 1, 2, \dots, n-1$. In downsampling, the band is aliased to frequencies $\left[0, \frac{f_s}{2n}\right]$. The aliasing is uninfluencial for any misinformation as the frequencies were filtered out by $H_m(f)$.

In the synthesis bank, the interpolation by factor k ($k = n$) replicates the subband $\left[\frac{mf_s}{2n}\right]$ at all subbands. Each subband is selected at its correct frequency range using synthesis bandpass filter $G_m(f)$, which has the same passband as $H_m(f)$.

Example In polyphase filtering, $K^t h$ - order FIR filter with transfer function H given by coefficients b such that Eq. 6.64 hold:

$$y[n] = \sum_{k=0}^K b[k]x[n-k] \quad (6.64)$$

Depending on number of samples, only one out of three groups will be unequal to zero. This is shown in Eq. 6.65

$$y[n] = \sum_{k=0}^{K_0} b[3k]x[n-3k] + \sum_{k=0}^{K_1} b[3k+1]x[n-3k-1] + \sum_{k=0}^{K_2} b[3k+2]x[n-3k-2] \quad (6.65)$$

Now if the output with different offsets separately, and keep only those inputs unequal to zero; the result consists of three sequences that are filtered versions of the signal before upsampling. These are written by Eqs. 6.66, 6.67, and 6.68.

$$y[n] = \sum_{k=0}^{K_0} b[3k]x[n-3k] \text{ is achieved by } H_0(z^3) \quad (6.66)$$

$$y[3n+1] = \sum_{k=0}^{K_1} b[3k+1]x[n-3k-1] \text{ is achieved by } H_1(z^3) \quad (6.67)$$

$$y[3n+2] = \sum_{k=0}^{K_2} b[3k+2]x[n-3k-2] \text{ is achieved by } H_2(z^3) \quad (6.68)$$

Regarding resolution transformation in multivariate signal processing, if 16 bits per sample are used, with 10 kHz sampling frequency that gives 160 kbits/sec, and dividing the signal into 2 bands, such as high frequency and low frequency subbands. High frequencies of speech are less important to intelligibility and therefore use only 8 bits per sample. The sample frequency can be reduced by a factor of 2 since bandwidth is halved, still satisfying Nyquist criterion. Thus $5 \times 16 + 5 \times 8 = 120$ kbits/s and thus the compression ratio is 160 : 120 which is 4 : 3. The reconstructed signal has no noticeable reduction in signal quality.

6.10 Quadrature Mirror Filter Bank (QMF)

The typical DFT based filter bank can not provide perfect reconstruction. The **Quadratic Mirror Filterbank** sub dividing a signal into octave band using critical sampling, where the octave is in the frequency range of $[f, 2f]$. In this filterbank, successive lowpass or highpass subdivisions into half bands decimation of the half bands by factor 2 after each subdivision lower band is recursively subdivided. The idea of simple two channel filter bank, $H_0(z)$ and $H_1(z)$ are high pass filters and

$L_0(z)$ and $L_1(z)$ are low pass filters. The idea of the QMF filter is filters $H_0(z)$, $H_1(z)$, $L_0(z)$, $L_1(z)$ and be designed so that aliasing in analysis part is eliminated in the synthesis part, and perfect reconstruction is achieved, even without processing of subbands.

In quadrature mirror filterbank, filter L is mirror image of H at $\pi/2$. H can be set as mirror of L by setting Eq. 6.69.

$$H(z) = L(-z) \quad (6.69)$$

6.11 Background Information

Estimation, detection, analysis are some common, yet compact representation of signals. Signals can be estimated using parametrical and non-parametrical methods. In parametric estimation, the signal is represented by a small number of parameters, then the task is to determine the model, and then model parameters (Douglas 1999). The non-parametric estimation of a signal is basis representations and expansions. Examples of parametric applications are Kalman filtering, particle filtering, and Wiener filtering. The Kalman filtering is the state representation of the best mean square estimation of some unknown input signal x , from the observation y (Wolfgang 2004). The Wiener filtering applies its stationary hypothesis for adaptive signal processing. The Kalman filter has been inefficient for some practical process whose distributions are non-Gaussian, multi-modal, or skewed, or, non-linear. For these, the extended Kalman filtering, and particle filtering has been proved to be very effective. One common approach for particle filtering is sample based pdf representation, where the posterior density of the particles of the pdf is characterized by Monte Carlo (MC) and importance sampling (IS) approach. The particles are representation by states, the MAP estimate of the particles in investigated, where the small importance particles are ignored, rather largest importance particles are collected. Simple short time spectral or STFT analysis which uses small signal components as segment and take DFT of the signal (Levy 2012; Stoica and Moses 2005). The STFT is a non-parametric estimation. The subband coding, filter bank analysis, Wavelet analysis are some examples of non-parametric approach. The subband coding, filter-bank are some multivariate signal processing tools. The signal is downsampled, analyzed applying different subband coding. The signal is up sampled and synthesized. Different filter bank and sub band coding approaches and stochastics analysis are discussed in DeGroot and Schervish (2012), Douglas (1999), Daumae (2017).

Readers can gain more information from references used in this chapter such as (Wolfgang 2004) and Daumae (2017)

6.12 Exercises

Exercise 1 Both zero mean, unit variance X and Y are two jointly Gaussian random variables such that $E[XY] = \rho$. Find:

1. MMSE estimate of X given Y ?
2. MVU estimate of X given Y ?
3. ML estimate of X given Y ?

Exercise 2 The recursive equations for the error and the filter coefficients of the least mean square algorithm are given by:

$$e(n) = d(n) - \hat{w}^H(n)u(n)$$

$$\hat{w}(n+1) = \mu u(n)$$

Write a function in Matlab, Python, C/C++, or Java which takes in input vector u and a reference signal d , both of length N , and calculates the error e .

Exercise 3 A FIR system has the following impulse response:

$$h(n) = \delta(n) + 1.6\delta(n-1) + 0.71\delta(n-2)$$

The input signal is $x(n)$ has 100 samples of unit variance of white Gaussian noise. Create the reference signal $d(n)$ by passing $x(n)$ through the filter.

1. Determine the range of values for step size μ so that the adaptive filter will be convergent in the mean.
2. Implement the adaptive filter using the given information.

References

- [Levy2012] Levy, R., Probabilistic Models in the Study of Language, Online Draft, Nov, 2012,
- [Stoica2005] Stoica, P, and Moses. R., Spectral Analysis of the Signals, Prentice Hall, 2005
- [DeGroot2012] DeGroot, H. M. and Schervish, M. J. , Probability and Statistics, 4th Edition, Addison-Wesley, 2012
- [Wolfgang2004] Wolfgang, K., Wavelets in Geodesy and Geodynamics, Walter De Gruyter, April, 2004
- [Daumae2017] Daumae, H., A Course in Machine Learning, CIML.info, 2017
- [Douglas1999] Douglas, S. C., chapter: Introduction to adaptive filters, Digital Signal Processing Handbook, CRC Press LLC, 1999

Chapter 7

Spectral Analysis



Overview

Basic spectral analysis is the quantitative distribution of the frequency information i.e., amplitude, correlation, and coherence. The correlation and coherence say the relations among more than one signal, and relation between the components or the frequency of the same signal. The most common frequency analysis technique is Fourier Transform (FT) discussed in Chap. 3. One of the popular spectral analyses is short time analysis discussed in Chap. 3 where the long random signal is segmented into small parts and these are used to apply the DFT. The spectral analysis of the real world random signals can be very complex. The spectral analysis is one of the primary steps to extract information for their classifications. The spectral analysis is also prior tasks to spectral estimation and signal modeling. The spectral analysis can be parametric and non-parametric. There are sub-band analysis, signal modeling based on estimation, prediction, and detection which also apply the probabilistic approach. In this chapter, we will first give a basic example of applying correlation and coherence, then this will be extended to analyze signals spectrally applying sub-bands and different types of signal modeling.

In this chapter, we also discuss multi-rate signal processing. In this processing, signals are analyzed at different sampling rates. In increasing and decreasing the sampling rate of the signal, interpolation and decimation are two basic operations of the multi-rate signal processing. For this we will also discuss about filter-bank that splits the signal in order to analyze and synthesize them.

7.1 Introduction

Signal processing relates measured data with theoretical concepts. Measured stochastic data by power spectral densities or spectral densities or autocorrelation functions assist in removing the redundant information. General estimation begins

with computing the mean, variance, and correlation between two variables. Observed random data may contain information that can be used to estimate unknown quantities such as mean, variance, and the correlation between two variables. The highest frequency component of the signal is the signal's bandwidth (BW). The signal is usually sampled at Nyquist rate, which is at least twice the highest frequency of the signal. When the sampling rate is just twice the highest frequency of the signal, then it is also called the critical sampling frequency or rate.

This chapter focuses on the transform domain adaptive filtering in the frequency domain. The transform domain adaptive filtering means the input signal is preprocessed by decomposing the input vector into orthogonal components, which are in turn used as inputs to a parallel bank of simpler adaptive sub band filters. With an orthogonal transformation, the adaptation takes place in the transform domain, as it is possible to show that the adjustable parameters are related to an equivalent set of time domain filter coefficients by means of the same transformation that is used for the real time processing.

7.2 Adaptive Spectral Analysis

Local Fourier Transform (LFT) and Wavelet Transform (WT) analyze signal variations and fluctuations with respect to time and scale. The local Fourier transform retains many of the characteristics of the usual Fourier transform with a localization given by the window function, which is thus constant at all frequencies. The WT decomposes in particular non-stationary signals into wavelets. The short time LFT uses a single analysis window and the WT uses short windows at high frequencies and long windows at low frequencies. Thus the WT does a local signal characterization better than the LFT.

7.3 Multivariate Signal Processing

Multivariate signal processing uses different sampling rates. This needs sampling rate conversion. Wavelets, Filterbanks, compression, coding are examples of multivariate signal processing methods. In multivariate signal processing, a signal is divided into multiple sampling rates, or subbands within the system. However, a change of sampling rate into some different sampling rates, requires the sampling rate conversion, such as upsampling and downsampling, then analyze them as subbands and then synthesize them in order to reconstruct the signal. The analysis is called an analysis bank and reconstruct the signal from M subband signals that are called a synthesis bank. This allows processing of each subband separately. Some common applications of multivariate signal processing in the real world problems are speech coding, image JPEG compression, and telecommunication. In Fig. 7.1,

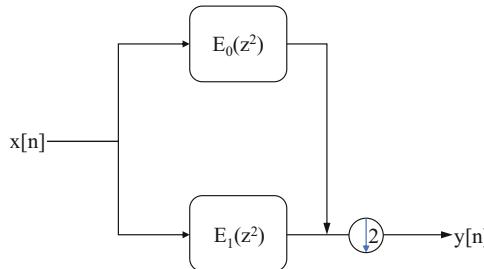


Fig. 7.1 Multivariate signal processing

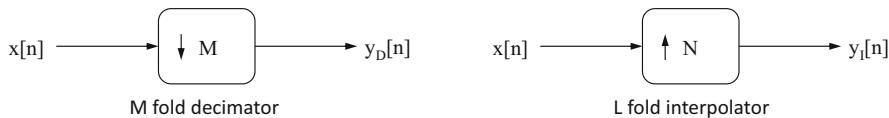


Fig. 7.2 Decimation and Interpolation

the signal $x[n]$ is divided into bands which is then decimated and downsampled by 2.

The problem of estimating power spectral density function $\varphi(\omega)$ of signal from a finite number of observations N is ill posed from a statistical standpoint, unless there is some approximations, estimations on the power spectrum density function. A basic operation of multivariate signal processing is to decompose a signal into a number of sub band components. In the processing, the signal is down sampled to the bandwidth of the frequency bands. Down sampling and up sampling are two fundamental parts of multivariate signal processing. The decimation in downsampling and interpolation in upsampling are shown in Fig. 7.2

In decimation, $M - 1$ samples are discarded keeping the M -th sample. While discarding $M - 1$ of every M input samples reduces the original sample rate by a factor of M , it also causes input frequencies above one-half the decimated sample rate to be aliased into the frequencies from DC to the decimated Nyquist frequency. The effect is compromised by applying a low pass filter to remove the components from portions of the output spectrum which are required to be alias free in subsequent signal processing steps. A benefit of the decimation process is that the lowpass filter may be designed to operate at the decimated sample rate, rather than the faster input sample rate by using an FIR filter structure and the output samples will discard $M - 1$ samples which will not be computed. If $x(n)$, $\forall n$, $0 \leq n \leq N - 1$ is an input signal, $h(k)$, $\forall k$, $0 \leq k \leq K$ are low pass filter coefficients, the output is $z(n)$ as shown in Eq. 7.1 before decimation.

$$z(n) = \sum_{k=0}^K h(k)x(n - k) \quad (7.1)$$

The output signal after decimation is $y(r) = z(rM)$, where the sampling rate is reduced by a factor M as shown in Eq. 7.2.

$$y(r) = \sum_{k=0}^K h(k) \times (rN - k) \quad (7.2)$$

In interpolation, $L - 1$ new samples are added in a uniform space, and each input sample increases the sample rate by a factor of L . While $L - 1$ new samples between each input sample increases the sample rate by a factor of L , it also introduces images of the input spectrum into the interpolated output spectrum at frequencies between the original Nyquist frequency and the higher interpolated Nyquist frequency. The effect is mitigated by filtering the interpolated signal with a low pass filter to remove any image frequencies which will disturb subsequent signal processing steps. A benefit of the interpolation process of using low pass filter is to operate at the input sample rate, rather than the faster output sample rate by using an FIR filter. Let $x(n)$ be the input signal, $v(n)$ is the sequence with $L - 1$ zeros inserted, $y(n)$ is the output sequence through $h(k)$ low pass filter with coefficients $h(0), h(1), \dots, h(k-1)$. The output sequence $y(n)$ is low pass filtered and is shown in Eq. 7.3 where $v(n-k) = 0$ unless $n-k$ is a multiple of L , the interpolation factor. This is because $L - 1$ zeros were inserted in the sequence $x(n)$ to get $v(n)$.

$$y(n) = \sum_{k=0}^K h(k)v(n - k) \quad (7.3)$$

In Eq. 7.4, the input signal $x(n)$ is filtered by $h(k)$ to generate interpolated signal $y(r)$.

$$y(r) = \sum_{n=0}^{\frac{K}{L}} h(r - Ln)x(n) \quad (7.4)$$

The reason of multivariate implementation rather than a direct time-domain version is related to computational time-domain version is related to computation complexity and convergence behavior. Since a filter bank computes a form of frequency analysis, subband adaptive filtering is a version of frequency domain adaptive filtering.

Example Design a decimator that downsamples input signal $x(n)$ by a factor $D = 2$. Use the Remez algorithm to determine the coefficients of the FIR filter that has 0.1 dB ripple in the passband and it is down by at least 30 dB in the stopband. Also determine the polyphase filter structure in a decimator realization that employs polyphase filters? A filter length $M = 30$ is given as the design specification, and the cut off frequency $\omega_r = \frac{\pi}{2}$.

The polyphase filter is obtained from $h(n)$, that have the impulse responses:

$$p_k = h(2n + k); \quad \forall k = 0, 1 \text{ and } n = 0, 1, 2, \dots, 14$$

Thus $p_0(n) = h(2n)$ and $p_1(n) = h(2n + 1)$. Hence one filter consists of the even numbered samples of $h(n)$ and the other filter consists of the odd-numbered samples of $h(n)$.

7.3.1 Sub-Band Coding and Subspace Analysis

Sub-bands analysis is used in many applications such as speech, image and video signal. In this analysis, the signal is first analyzed and synthesized. In the analysis, signal is decomposed and decimated. In the synthesis stage, signal is interpolated and reconstructed. There are different techniques and methods to analyze the signals spectrally. Some of these techniques are filter bank where the combination of the filter banks can be of different type, based on application, construction of such as perfect or not perfect construction, tree typed combination such as quadrature mirror typed filter bank, transform-domain type, adaptive sub-band filtering type. The adaptive filter bank type can be frequency domain adaptive filter such as STFT and sub-band adaptive filter such as quadrature mirror filter type. In Fig. 7.3, we see a typical multivariate signal processing. In Fig. 7.3 his signal is subdivided into some sub-bands. These are downsampling in the analysis stage. This is labelled

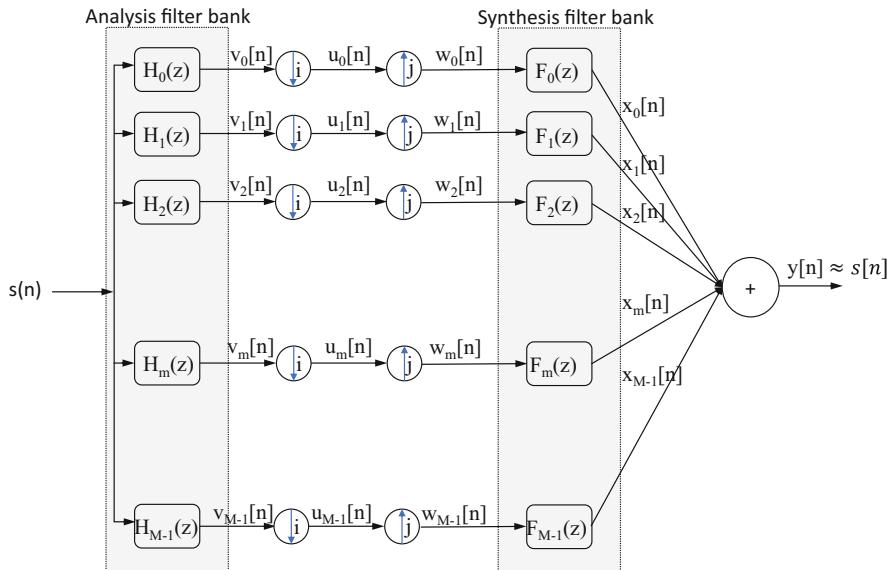


Fig. 7.3 Sub-band coding

as analysis filter bank and the subbands are synthesized, and reconstructed at the synthesis stage. This is called the synthesis filter bank. In this figure, i and j are representing the quantities for downsampling and for upsampling.

In MPEG audio coding, a psychoacoustic model is used to decide how much quantization error can be tolerated in each sub-band, while signals below the hearing threshold of a human listener are discarded. In the sub-bands, that can tolerate more error, hence less bits are used for coding. The quantized subband signals can then be decoded and recombined to reconstruct an approximate version of the input signal. Such processing allows on average 12 to 1 reduction in bit rate while still maintaining CD quality audio. The psychoacoustic model takes into account the spectral masking phenomenon of the human ear, which says that high energy in one spectral region will limit the ear's ability to hear details in nearby spectral regions. Therefore, when the energy in one sub-band is high, nearby subbands can be coded with less bits without degrading the perceived quality of the audio signal.

Example Design a filter band with a digital FIR filters for a speech application to split a total bandwidth of 8 kHz with equal bandwidth sub-bands.

Solution Suppose the sampling frequency is 8 kHz and the set of uniformly band pass filters. The filter points for the four band-pass filters are 0 and 1 kHz, 1 to 2 kHz and 3 kHz to 4 kHz. First design a low pass filter with a bandwidth of 1 KHz, at a sampling rate of f_s of 8 kHz, a frequency of 1 kHz corresponds to a normalized angular frequency of $\omega_n = 2\pi \frac{f}{f_s} = \frac{\pi}{4}$ radians and the normalized frequency of $f_n = \frac{f}{f_s} = \frac{1}{8} = 0.124$. By using window design technique, the FIR impulse response is obtained by taking the inverse Fourier transform as:

$$h_d[m] = \int_{-0.125}^{+0.125} 1.0e^{j2m\pi f} df \quad (7.5)$$

7.4 Wavelet Analysis

The wavelet analysis calculates the correlation between the signal under consideration and a wavelet function $\varphi(t)$. The similarity between the signal and the analyzing wavelet function is computed separately for different time interval, resulting in a two dimensional representation. The analyzing wavelet function $\varphi(t)$ is also referred to as the mother wavelet. Compare to Fourier transform, the analyzing function of the wavelet transform can be chosen with more freedom, without the need of using sine-forms. A wavelet function is a small wave, which must be oscillatory in some way to discriminate between frequencies. The wavelet $\varphi(t)$ contains both the analyzing shape and the window. An analyzing function is classified as wavelet $\varphi(t)$ if the following mathematical criteria are satisfied:

1. A wavelet must have finite energy.

$$E = \int_{-\infty}^{+\infty} |\varphi(t)|^2 dt < \infty \quad (7.6)$$

The energy E equals the integrated squared magnitude of the analysing function $\varphi(t)$ and must be less than infinity.

2. If $\psi(f)$ is the Fourier transform of the wavelet $\varphi(t)$, the following condition must hold.

$$C_\varphi = \int_0^{+\infty} \frac{|\hat{\varphi}(f)|^2}{f} df < \infty \quad (7.7)$$

This condition implies that the wavelet has no zero frequency component $\psi(0) (= 0)$, i.e. the mean of the wavelet $\psi(f)$ must equal to zero. This condition is known as admissibility constant. The value of C_φ depends on the chosen wavelet.

3. For complex wavelets the Fourier transform $\psi(f)$ must be both real and vanish for negative frequencies. The transformation of the signal $x(t)$ applying the wavelet $\varphi(t)$ is

$$X_{(\tau,s)} = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t) \varphi\left(\frac{t-\tau}{s}\right) dt \quad (7.8)$$

The $X_{(\tau,s)}$ is a function of the translation parameter τ and the scale s . The signal energy is normalized at every scale by dividing the wavelet coefficients ($\frac{1}{\sqrt{|s|}}$). This ensures that the wavelets have the same energy at every scale. The mother wavelet is contracted and dilated by changing the scale parameter s . The variation in scale changes not only the central frequency f_c of the wavelet, but also the window length. Therefore, the scale s is used instead of the frequency for representing the results of the wavelet analysis. The translation parameter τ specifies the location of the wavelet in time, by changing τ the wavelet can be shifted over the signal. For constant scale s and keeping the translation τ constant fills the columns of the time-scale plane. The elements in $X_{(\tau,s)}$ are called wavelet coefficients, each wavelet coefficient is associated to a scale i.e. frequency and a point in the time domain. The inverse transform of the wavelet is

$$x(t) = \frac{1}{C_\varphi^2} \int_{-\infty}^{+\infty} X_{(\tau,s)} \frac{1}{s^2} \varphi\left(\frac{t-\tau}{s}\right) d\tau ds \quad (7.9)$$

The admissibility constant C_φ must satisfy the second wavelet condition. A wavelet function has its own central frequency f_c at each scale, the scale s is inversely proportional to that frequency. A large scale corresponds to a low frequency, giving global information of the signal. Small scales correspond to high frequencies, providing detail signal information.

Using scales and shifts of a prototype wavelet, a linear expansion of a signal is obtained. The scales used are powers of an elementary scale factor (typically 2), the analysis uses a constant relative bandwidth i.e. the frequency axis is logarithmic. The sampling of the time-frequency place is now very different from the rectangular grid used in the STFT. The lower frequencies where the bandwidth is narrow, i.e. the basic functions are stretched in time, are sampled with a large time step, while high frequencies, which correspond to short basis functions, are sampled more often. The wavelet analysis allows good orthonormal bases, where STFT does not. In block transformation, the signal is locally analyzed using a block transform, where the sequence is segmented into adjacent blocks of N samples, and each block is individually transformed. The filter banks can opt with both the STFT and wavelet like analysis, which is constant relative bandwidth in frequency.

The wavelet transform analyses the signals using wavelets, where the Fourier transform analyses the signal in terms of a set of trigonometric basis functions such as sine and cosine functions. The wavelets are in the form of translations and dilations of a mother wavelet. The wavelets are localized in time and frequencies. To improve the sparsity of complicated sounds or images, one can consider several orthogonal bases that compose a large dictionary of atoms. In this framework, one has to choose a best basis adapted to the signal to process. To enable fast computation, this dictionary is required to have a tree structure, so that the best basis can be optimized using a fast dynamic algorithm.

The subband coding, filterbanks, quadrature mirror filters are the techniques developed based on wavelet analysis. From the mathematical point of view, a filter bank carries a series expansion where the subband signals are coefficients, and the time-shifted variants $f_k(n - iN)$, $i \in Z$, of the synthesis filter impulse responses $f_k(n)$ form the basis. The main difference to the block is that the lengths of the filter impulse responses are usually larger than N , so that the basis sequences overlap. Examples: In Fig. 7.2, the sequences are formulated using Eqs. 7.5, 7.6, 7.7, 7.8, and 7.9.

$$\{x[n]\} = \{2, 6, 4, 8\} \quad (7.10)$$

$$x_0[n] = \frac{1}{2}(x[n] + x[n - 1]) \quad (7.11)$$

$$x_1[n] = \frac{1}{2}(x[n] - x[n - 1]) \quad (7.12)$$

$$\{x_1(n)\} = 1, 4, 5, 6, 4 \quad (7.13)$$

$$\{x_2(n)\} = 1, 2, -1, 2, -4 \quad (7.14)$$

Down-sampling gives the sub band components:

$$\{x_{D_1}(n)\} = \{1, 5, 4\} \quad (7.15)$$

$$\{x_{D_2}(n)\} = \{1, -1, -4\} \quad (7.16)$$

For reconstruction, the signals $\{x_{D_1}(n)\}$ and $\{x_{D_2}(n)\}$ are up-sampled such that

$$\{v_1(n)\} = \{1, 0, 5, 0, 4, 0\} \quad (7.17)$$

$$\{v_2(n)\} = \{1, 0, -1, 0, -4, 0\} \quad (7.18)$$

Finally, the filters

$$G_1(z) = 1 + z^{-1} \quad (7.19)$$

and

$$G_2(z) = -1 + z^{-1} \quad (7.20)$$

gives

$$x_{E_1}(z) = v_1(n) + v_1(n-1) \quad (7.21)$$

$$x_{E_2}(z) = -v_2(n) + v_2(n-1) \quad (7.22)$$

Thus

$$\{x_{E_1}(n)\} = \{1, 1, 5, 5, 4, 4\} \quad (7.23)$$

$$\{x_{E_2}(n)\} = \{-1, 1, 1, -1, 4, -4\} \quad (7.24)$$

The reconstructed signal

$$y(n) = x_{E_1}(n) + x_{E_2}(n) \quad (7.25)$$

Thus

$$\{y(n)\} = \{0, 2, 6, 4, 8, 0\} \quad (7.26)$$

The $\{y(n)\}$ is the delayed version of the original signal $\{x(n)\}$ (Fig. 7.4).

7.5 Adaptive Beam Forming

The basic concept of an adaptive beam forming is wave propagation phase relationships. It focuses on maximizing the signal to noise ratio. If the input array signal vector is x as shown in Eq. 7.27, then the adaptive beamforming occurs if

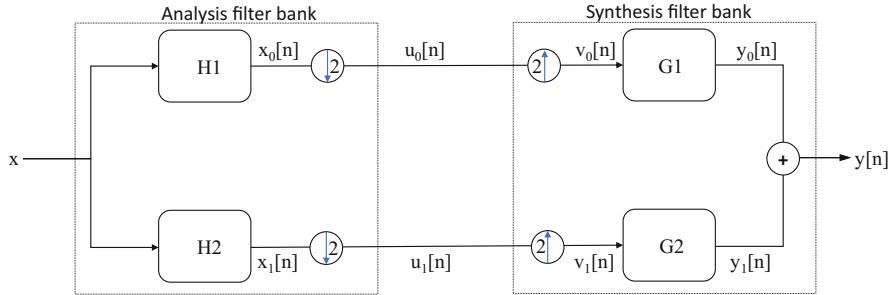


Fig. 7.4 Basic Structure of a filter bank

w_1, w_2, \dots, w_N are determined and optimized based on the input and output array signals. If the array signal vector is:

$$x(i) = [x_1(i) x_2(i) \dots x_N(i)] \quad (7.27)$$

then the beam-former output is shown in Eq. 7.28 is:

$$x(k) = [x_1(k) x_2(k) \dots x_N(k)] \quad (7.28)$$

The beam-former outputs are formulated by the following Eqs. 7.29 and 7.30.

$$y(k) = w^H x(k) \quad (7.29)$$

$$x(k) = x_s(k) + x_N(k) + \dots + x_I(k) \quad (7.30)$$

In Eq. 7.30, $x_s(k)$ is the signal, $x_N(k)$ is the noise, and $x_I(k)$ is interfering signal. The goal is to filter out x_I and x_N as much as possible, therefore obtaining an approximation \hat{x}_s of actual signal x_s . The most popular criteria of adaptive beam-forming is:

- Mean Squared Error (MSE) minimum:

$$\min_w MSE, \quad MSE = E\{|d(i) - w^H x(i)|^2\} \quad (7.31)$$

- Signal to Interference plus Noise Ratio (SINR):

$$\max_w SINR, \quad SINR = \frac{E\{|w^H x_s|^2\}}{E\{|w^H (x_I + x_N)|^2\}} \quad (7.32)$$

In case of $\max SINR$ criterion,

$$x(k) = s(k)a_s + x_I(k) + x_N(k) \quad (7.33)$$

In Eq. 7.33, a_s is known steering vector of the desired signal and then,

$$SINR = \frac{\sigma_s^2 |w^H a_s|^2}{w^H E\{(x_I + x_N)(x_I + x_N)^H\}w} = \frac{\sigma_s^2 |w^H a_s|^2}{w^H R w} \quad (7.34)$$

In Eq. 7.34

$$R = E\{(x_I + x_N)(x_I + x_N)^H\} \quad (7.35)$$

Equation 7.35 is the interference plus noise co-variance matrix. The $SINR$ does not depend on re-scaling of w , i.e. if w_{opt} is an optimal weight, then αw_{opt} is such a vector too. Therefore, $\max SINR$ is equivalent to $\min_w w^H R w$ subject to:

$$w^H a_s = \text{constant} \quad (7.36)$$

If the constant is 1, then

$$H(w) = w^H R w + \lambda(1 - w^H a_s) + \lambda^*(1 - w^H a_s) \quad (7.37)$$

$$\nabla_w H(w) = (R w - \lambda a_s)^* = 0 \quad (7.38)$$

$$R w = \lambda a_s \Rightarrow w_{opt} = \lambda R^{-1} a_s \quad (7.39)$$

Presented Eq. 7.39 is a special version of Wiener-Hopf equation. From the constrain equation, we obtain:

$$\lambda = \frac{1}{a_s^H R^{-1} a_s} \quad (7.40)$$

Therefore, this can be written by Eq. 7.41 presented next:

$$w_{opt} = \frac{1}{a_s^H R^{-1} a_s} R^{-1} a_s \quad (7.41)$$

Substituting w_{opt} into SINR expression, we obtain Eq. 7.42

$$SINR = SINR_{opt} = \frac{\sigma_s^2 (a_s^H R^{-1} a_s)^2}{a_s^H R^{-1} R R^{-1} a_s} = \sigma_s^2 a_s^H E^{-1} a_s \quad (7.42)$$

If there are no interference sources i.e. only white noise with variance σ^2 formulated in equation below 7.43:

$$SINR_{opt} = \frac{\sigma_s^2}{\sigma^2} a_s^H a_s = \frac{N \sigma_s^2}{\sigma^2} \quad (7.43)$$

The optimal SINR is happened to be if the co-variance matrix includes the signal component formulated by Eq. 7.44 presented next.

$$R_x = E\{xx^H\} = R + \sigma_s^2 a_s a_s^H \quad (7.44)$$

Using the matrix inversion lemma, we arrive in Eq. 7.45

$$\begin{aligned} R_x^{-1} a_s &= (R + \sigma_s^2 a_s a_s^H)^{-1} a_s = \left(R^{-1} - \frac{R^{-1} a_s a_s^H R^{-1}}{\frac{1}{\sigma_s^2} + a_s^H R^{-1} a_s} \right) a_s \\ &= \left(1 - \frac{a_s a_s^H R^{-1}}{\frac{1}{\sigma_s^2} + a_s^H R^{-1} a_s} \right) R^{-1} a_s = \alpha R^{-1} a_s \end{aligned} \quad (7.45)$$

Equation 7.45 holds only if there is an infinite number of snapshots and a_s is known exactly. Gradient algorithm maximizing SNR which is very similar to LMS as written in Eq. 7.46 next:

$$w_{k+1} = w_k + \mu (a_s - x_k x_k^H w_k) \quad (7.46)$$

There, again, we use the simplifying notation as $w_k = w(k)$ and $x_k = x(k)$. The vector w_k converges to $w_{opt} \approx R^{-1} a_s$ if Eq. 7.47, given below, is true:

$$0 < \mu < \frac{2}{\lambda_{max}} \Rightarrow 0 < \mu < \frac{2}{tr\{R\}} \quad (7.47)$$

The disadvantage of the gradient algorithms is that the convergence may be very slow, i.e. it depends on the **Eigen** values spread of R .

7.6 Independent Component Analysis (ICA)

Independent Component Analysis is a very powerful technique, which is able to separate independent sources linearly mixed in several samples. The ICA also a useful tool to separate artefacts embedded in the data. The ICA estimates the independence and non-gaussianity of the mixed process originating from some different objects or sources. A common application of ICA is cocktail party problem. An illustration of such problem is a number of speakers speaking to a number of microphones at a party where the speech are mixed with one another but one wants to find the independent or separation of the speeches spoken by each speaker or one wants to obtain a specific speech spoken to a specific microphone placed in the party where the spoken speech are mixed and each microphone has the recorded mixed speech. The solution of the problem of separation of the speech is obtained applying ICA. To formalize the problem in Eq. 7.48, A is an unknown

square matrix called the mixing matrix, observation is $x^{(i)}; i = 1, 2, \dots, m$, the goal is to separate the mixed sources $s^{(i)}$.

$$x = As \quad (7.48)$$

The Eq. 7.48 can be rewritten as Eq. 7.49 given next:

$$x^{(i)} = As^{(i)} \quad (7.49)$$

where $s^{(i)}$ is an n dimensional vector, $s_j^{(i)}$ is the speech spoken by a person denoted by j at time i . Similarly, $x^{(i)}$ is an n -dimensional vector, and $x_j^{(i)}$ is recorded speech spoken by j at time i . If $W = A^{-1}$ is the un-mixing matrix, we rewrite Eq. 7.49 as Eq. 7.50:

$$s^{(i)} = Wx^{(i)} \quad (7.50)$$

For j th source, Eq. 7.50 turns into Eq. 7.51. R in the equation indicates the real numbers and $W = [-w_1^T, -w_2^T, \dots, -w_n^T]^T$ where w_i^T denotes the i -th row of W .

$$s^{(i)} = W_j^T x^{(i)} \quad \text{for } w_i \in R^n \quad (7.51)$$

The PCA is based on second order statistics and the data fit Gaussian distribution i.e. the correlation and covariance properties. If the data is not characterized by the second order moment, than ICA is useful. The PCA has a wide range of applications when the data is Gaussian, linear, and stationary. But if the data can not be defined by Gaussian distribution, i.e. non-Gaussian via central limit theorem, when the raw noisy data, when a sensor records several source signals simultaneously, then ICA plays a significant role.

The ICA thrives on the fact that the data are non-Gaussian. This implies that the ICA exploits the loose end of the central limit theorem which states that the distribution of a sum of independent random variables tends toward a Gaussian distribution. Fortunately, for ICA, there are many cases where some real-world data do not have sufficient data pools that can be characterized as Gaussian.

There is also the fast ICA which uses negentropy¹ concept as follows:

1. First, in the fast ICA, the data \mathbf{x} is concentrated such as zero mean as written by Eq. 7.52

$$\mathbf{x} = x - x_m \quad x_m = E\{x\} \quad (7.52)$$

2. Then the x is maximized non-Gaussian characteristics, i.e. PCA with filtering can be formulated by Eq. 7.53

¹ a measure of distance to normality.

$$z = V \wedge^{-\frac{1}{2}} V^T x, \quad V \wedge V^T = E\{xx^T\} \quad (7.53)$$

3. Choose an initial random vector, w , $\|w\| = 1$
4. Update w , (maximally non-Gaussian director) using Eqs. 7.54, 7.55, and 7.56:

$$w = E\{z \times g(w^T z)\} - E\{g'(w^T z)\}w \quad (7.54)$$

$$g(y) = \tanh(a_1 y) = y \times \exp\left(-\frac{y^2}{2}\right), \quad 1 < a_1 < 3 \quad (7.55)$$

$$w = \frac{w}{\|w\|} \quad (7.56)$$

5. If not converged, then go back to step 4.
6. Obtain the Independent component, s using Eq. 7.57 following step 7.

$$s = [w_1, w_2, \dots, w_n]x \quad (7.57)$$

ISA is used to estimate the sources given noise measurements. In the following examples, we see ICA separates three different sources which are collected by three different microphones and sources. The sources of the sound are three different kinds of instrumental sound in a musical concert. In Fig. 7.5, we see how the ICA estimates sources from noisy measurements. In this case, the mixed signal, namely the musical audio source, is mixed with three simultaneous musical instruments. The mixed signal is recorded using three microphones. How the ICA separated the source is shown in details in Fig. 7.5 https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_blind_source_separation.html#sphx-glr-auto-examples-decomposition-plot-ica-blind-source-separation-py. In the following example, the PCA fails to recover the instrumental sounds as they are not truly Gaussian process, but rather non-Gaussian.

7.7 Principal Component Analysis (PCA)

The **Principal Component Analysis** identifies the principle directions in which the data varies. For a given set of data, the principle component analysis finds the axis system defined by the principle directions of variance. If the variation in a data set is caused by some natural property such as an experimental error, than it is normally assumed to be normally distributed. In such case, the nominal extent of the data is hyper-ellipse. This encloses the data points to a class assuming the data points belong to the class. If the variation of the data caused by some other reasons, the PCA reduces the dimensionality of a data set.

The principle components are obtained by calculating the eigenvectors and eigenvalues of the data covariance matrix. The process is equivalent to finding the

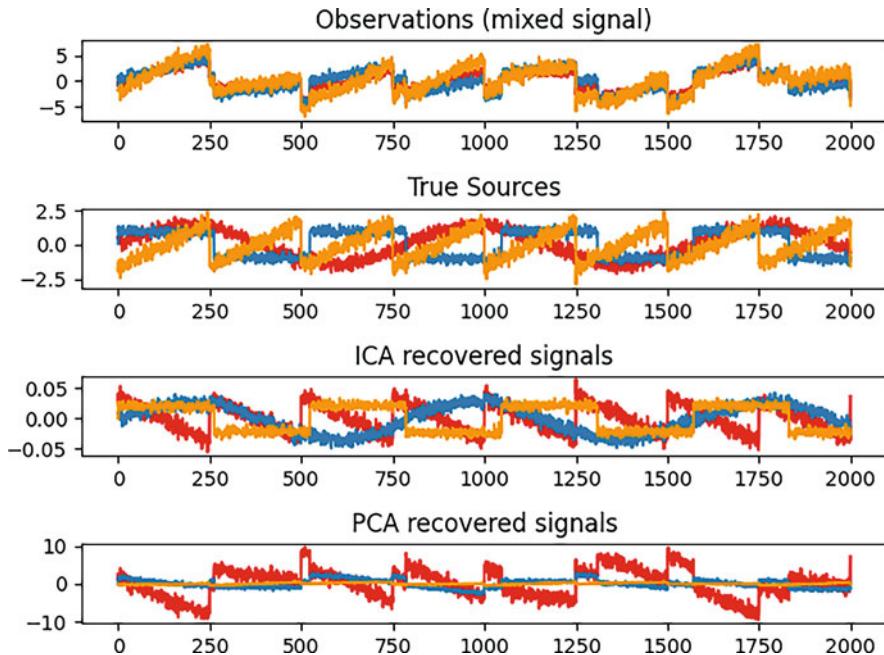


Fig. 7.5 ICA separates the mixtures into its three different sources

axis system in which the covariance matrix is diagonal. The eigenvector with the largest eigenvalue is the direction of the greatest variation, the one with the second largest eigenvalue is the orthogonal direction with the next highest variation. The principle components decomposition of X can be written in Eq. 7.58, where Y is a $n \times n$ matrix whose columns are the eigen vectors of $X^T X$.

$$P = XY \quad (7.58)$$

Let γ be an eigen value of X , then there exists a vector l such that Eq. 7.59 holds:

$$AX = \gamma X \quad (7.59)$$

The vector X is called an eigen vector of A associated with the eigen value *gamma*. The difference between PCA and ICA is the PCA minimizes the covariance of the data, on the other hand ICA minimizes the higher order statistics such as fourth-order cumulant or kurtosis, thus minimizing the mutual information of the output. The PCA yields orthogonal vectors of high energy contents in terms of the variance of the signals, whereas ICA identifies independent components for non-Gaussian signals. ICA thus possesses two ambiguities: First the ICA model equation is underdetermined system; one can not determine the variances of the independent components. Second, one can not rank the order of dominant components.

In typical pattern recognition problems, a PCA is often followed by a multiple discriminant analysis (MDA). Both multiple discriminant analysis (MDA) and the principle component analysis are linear transformation methods and closely related to each other. In PCA, the desired information is to find the directions that maximize the variance in the dataset, whereas in MDA it is to find the directions that maximize the separation or discrimination between different classes, for example, in pattern classification problems where the dataset consists of multiple classes. In PCA, the class labels are ignored. In other words, in PCA, the focus is the projections of entire dataset without class labels on a different subspace, and in MDA, the focus is to determine a suitable subspace to distinguish between patterns that belong to different classes. In other words, the PCA tries to find the axes with maximum variances where the data is most spread within a class, since the PCA treats the whole data set as one class and the MDA maximizes the spread between classes.

7.8 Best Basis Algorithms

For wavelet packets, selection is the predominant mode of operation. Basis averaging could be considered but with a little success with respect to its practical application. One approach is to select the whole set of wavelet packet coefficients which can be rapidly computed in $O(N \log(N))$ operations. One common approach to select threshold is to select a basis first, then threshold, or vice-versa. This has also not much use in practice. One may use Coifman-Wickerhauser best basis algorithm approach i.e. the best basis method for signal compression. This considers a basis that gives the most efficient representation of a signal. Here efficient refers the optimal sparse. A vector of coefficients is said to be sparse if most of its entries are zero, and only a few are non-zero. The Shannon entropy suggested as a measure of sparsity. Given a set of best basis $\{v_i\}$, the Shannon entropy can be written as $\sum |v_i|^2 \log |v_i|^2$.

The Shannon entropy can be used to measure the sparsity of a vector, and the Coifman-Wickerhauser algorithm searches for the basis that minimizes the overall negative Shannon entropy which computes the general cost functions. Coifman and Wickerhauser show that the finest basis can be obtained by starting from the finest scale functions and comparing the entropy of that representation by the next coarsest scale packets, and then selecting the one that minimizes the entropy either the packet or the combination of two children. Then this operation is applied recursively if required.

7.9 Background Information

Spectral analysis is the analysis of the signal's frequency in the frequency domain. In mathematics, the spectral analysis of signals is the Eigen value and Eigen vector decomposition. The periodogram, the correlogram, power spectrum density

(PSD) are some basic spectral analysis tools. They are developed based on weak conditions. In some common scenario, statistical signals are analyzed using some form of stationarity, they can be weak sense stationarity (WSS), or wide sense stationarity (WSS), and strict sense stationarity (SSS). The weak sense stationarity concerns with shift-invariance in time of first and second moments which are mean and covariance functions are time invariant. The strong sense stationarity (SSS) maintains the WSS criteria, and additional concerns which is the second order joint pdf depends only on the time difference of the consecutives. In simple terms, the spectral analysis of time series analysis is the decomposition of the time series signal into a sum of sinusoidal components with unrelated coefficients. The spectral analysis can be parametric, and non-parametric. In parametric spectral analysis, the signal is parameterized by a model, and parameters are extracted from signals, also the parameters represent the signals. In non-parametric signal modeling, the signal is analyzed by the basis representations and expansion (Hastie et al. 2001). Parametric spectral analysis is one of the basic spectral analyses in adaptive signal processing, where the parameters are extracted by some signal model, and the parameters represent the signal (Vaseghi 2006). Examples of some non-parametric spectral analysis are the analysis of signal applying filter bank theory, wavelet transform, conjugate of wavelet transformation, which uses local trigonometric transformations to analyze the signals spectrally. There are some adaptive spectral analysis which uses some optimization approach. Examples of some analysis are ICA, PCA, BSS. These are briefly introduced in this chapter (Hyvärinen et al. 2001).

7.10 Exercises

Exercise 1 Given an image, implement PCA whitening.

Exercise 2 Write an ICA module, and apply this to the matrix, where each row represents a mixed signal from four different source signals.

References

- [Vaseghi2006] Vaseghi, S.V., Advanced Digital Signal Processing and Noise Reduction, Wiley, 3rd Ed., January, 2006
- [Hyvarinen2001] Hyvärinen, A., Karhunen, J., Oja, E., Independent Componenet Analysys, Wiley, 2001
- [Hastie2001] Hastie, T., Tibshirani, R., Friedman, J., The Elements of Statistical Learning. New York: Springer, 2001

Part II

Machine Learning and Recognition

Learning is a process. In the learning process one formulates how the data are organized in order to give the best learning outcome. The outcome is approximated step by step and there may be no termination of the learning process. In fact, there may not even be a precise description of the learning goal.

This part elaborates general aspects and principles of learning. An important aspect is the improvement of the learner after learning. In this part we emphasize on elements related to stochastic processes. Stochastic processes do not provide directly the information in which one is interested. For extracting that information, learning methods are developed and used. This is a central part of Part II in this book. Learning the data patterns and parameters is the common method for extracting the desired information about this data. Through learning, the data is understood and characterized by machines.

Available data are split into training data and test data. The training data are used for approaching the desired result and the test data for checking the quality of the result. The training data has to resemble the test data. This resemblance determine the success of the machine learning. Sometimes training data itself is further split for using a fraction of it as cross-validation, namely for detecting and preventing over-fitting phenomena.

This section discusses stochastic processes, and how the signals are analyzed for learning and recognition.

Generalities About Part II

Space with its elements are denoted as states. These states describe the situations in which the signals happen. The number of states is N and one denotes these states by $Q = f_{q_1}, f_{q_2}, \dots, f_{q_N}$. The actual states are in general unknown in each situation, i.e. after the observation. In addition one has observations that are known. The states are not completely unknown: One has probabilities of the states given the observations. Markov processes, Hidden Stochastic Processes (HSM) and Mixed

Markov Processes (GMM). In HSM the probabilities are unknown and the GMM contain a mix of different Gaussian probabilities. Much interest is put on recognition and search, in particular Bayesian reasoning.

Overview of Part II

Part II addresses the ideas of learning with a particular insight into the influence of the learning goals. Learning is explained as a search of parameters with which the signal generator would best fit measured observations. Methods are exemplified for learning the parameters of models used in representing relations between communication intentions and speech signals. These methods constitute the foundation of technologies for speech recognition.

Principled representation of uncertainty about system states that cannot be measured can be based on joint probabilities. Efficiency of representation and computation is achieved by explicitly exploiting domain knowledge about conditional independence in Bayesian Networks and in their pattern specialized in modeling discrete time under Dynamic Bayesian Networks. Hidden Stochastic Models are methods of using Dynamic Bayesian Networks to represent the processes generating signals when parts of them are not measurable. Integration of utility as way of specifying goals into signal generation models and processing is the basis of optimization in learning.

Learning frequently exploits methods of pre-processing signals for extracting features to reduce dimensionality of data and thereby to reduce complexity of general classifier. One of the main methods of extracting features is based on Linear Prediction Coefficients which is in itself a process of learning signal models. Estimation of hidden states and of probabilities of matching models to models is also presented as learning processes with particular goals, yielding the Baum-Welch algorithm ideas.

Unsupervised Learning is a family of techniques extensively used in reducing data dimensionality in view of supporting classification tasks. The examples of clustering techniques are shown with their immediate applications to vector quantization and Gaussian mixture models. The expectation maximization procedure for training GMMs is also presented in this context.

The part proceeds with detailed presentation of applications aggregating several of the presented techniques such as in Hidden Markov Model based speech recognition. Applications are also shown for specific learning concepts for representing uncertainty in ambiguity of natural speech with rough sets and fuzzy set. A chapter is dedicated to one of the most pervasive learning technology of our times, the neural network.

Main Topics in Part II

This chapter presents the technology and applications of the learning for signal processes. It covers some specific aspects about learning, such as:

- Supervised learning
- Unsupervised learning, and
- Semi-supervised learning
- Goals, utility and search
- Bayesian networks and Hidden Stochastic Models
- Linear Features and Linear Prediction Coefficients
- K-means, GMMs, EM, Hidden Markov Models
- Fuzzy logic and rough sets
- Neural networks.

Chapter 8

General Learning



Overview

Learning is a process of gaining knowledge from experience and adapting the behavior of a system to the encountered environment based on the thus obtained knowledge. Machine learning is commonly concerned with designing computer models of real processes and training these models from large data sets. In applications there is no ideal model but repeated learning, experimentation, and model improvement.

Learning requires to apply logic, knowledge, experience, reasoning, and mathematical methods in order to perceive, model, compute, and adapt to new situations. This chapter discusses the essence of learning, and related concepts from machine learning, artificial intelligence, applied math and science for engineering.

8.1 Introduction to Learning

A general visualization of the learning process is shown in Fig. 8.1. The “Learner” with input from the “Teacher” and “Special Information”, which is integrated using selected “Criteria”, changes into a new version of itself. Figure 8.1 also shows some optional components of the learning process. On the other hand, additional elements may come in. The main aspect the figure shows is that the learning changes the state of the learner. In the changed state the learner will act differently and this is the purpose of learning.

The learner is influenced by various outside constraints and data sources as decided by the learning system. The constraints are required as they give the learner desired abilities, such as the possibility to be stored in given computer memory, and to respect known or desired properties. If the content of the memory holding the learner state is changed then the behavior of the learner will be different.

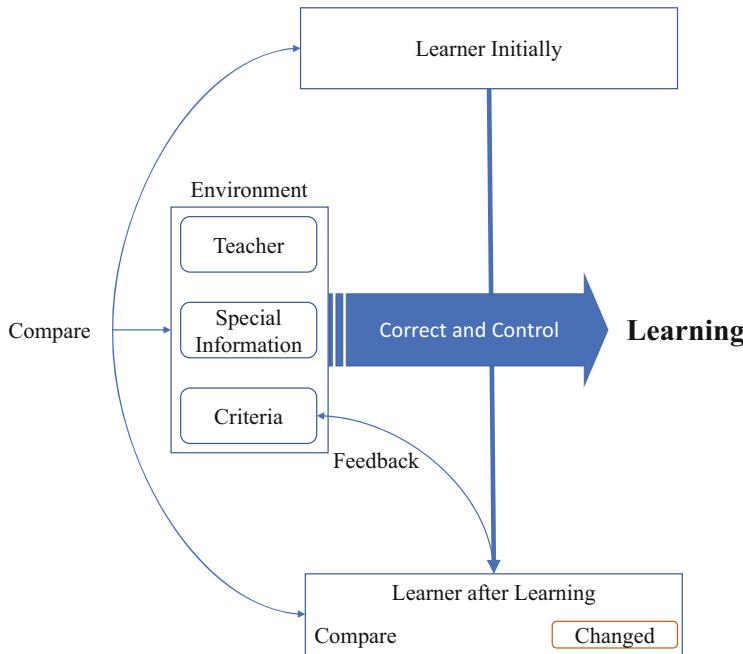


Fig. 8.1 Learning

For a set of targeted processes the main goal of machine learning is to reveal properties and parameters of interest that are not explicitly available but rather hidden in given examples. The goal is making properties easily visible or exploitable via the obtained representation of the acquired knowledge into the learner memory.

Learning is an interdisciplinary activity that covers many research areas. A perspective discussed here comes from machine learning that is in itself a quite large area. The learner is an agent that in our case can be seen as a machine. If it is a machine then it should have rules for learning. Learning is itself a process and has a sender and a receiver. The assumption is that the receiving agent has a behavioral change because of the received message. The memory is a quite essential element of the learner. The hope is that the change from learning results in a better behavior of the learner. One type of problem of interest in this book is machine learning in the presence of stochastic processes. Learning is always focused on finding parameters of a constrained model, connected with the improvement of the behavior of the learner.

While from the available information and data one can learn many things, what is really learned depends on the interest of the learner. Learning is goal directed and needs a strategy.

For learning processes one distinguishes two phases, the training phase and the test phase. In the test phase one controls the learned objects for accuracy and usefulness. In Fig. 8.2 the two phases are distinguished: training and testing. A

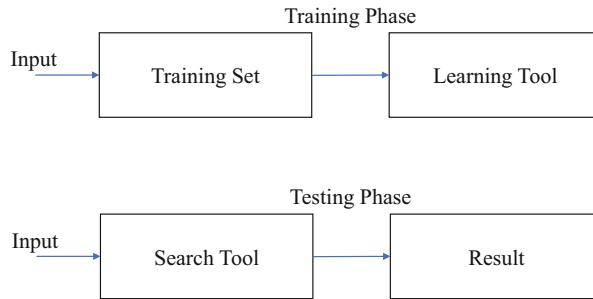


Fig. 8.2 The learning phases

search is supposed to select from among the possible training results by using the evaluations offered by the tests.

8.2 The Learning Phases

The learning phase has its sub-phases and each learning sub-phase has different parts.

First one has to gather the examples that are representative for what one wants to learn. Then the proper learning starts: this is called the training phase. In this phase one slowly increases the knowledge and optimizes behavior by adapting the learned elements (e.g., by back-propagation in artificial neural network: work by Rumelhart, Hinton, and Williams, titled “Learning Representations by Back-Propagating Errors”). Since there are always many possibilities for adaptation one has to search for one satisfying target criteria. Therefore the set of criteria has to be first established. Then, one has to evaluate the training results using test data. A consequence is that one keeps certain learned results that pass the control criteria while other results are rejected.

The learning phases commonly employ two sets of examples:

- Examples for learning
- Examples for testing.

A detailed application is provided in the next chapter.

8.2.1 *Search and Utility*

After learning took place there is still the problem of using the results in practice. It is expected that applications get advantages from the learning.

In contrast to logical reasoning, learning provides no immediate results. Sometimes learned models are worse compared with the model available before learning. It is the purpose of the control component to discover this.

Practically the control decides to forget certain learned aspects obtained from the training methods. Often there is no exact theory describing what is a better behavior. When properties are discovered by learning, they are not necessarily true. In certain situations the properties are completely determined. However, in many situations an uncertain information is sufficient for the user.

Up to now the question of when one should keep something learned, is still open. Intuitively, one would keep something that is useful. Utility functions are also constructed according to the local-global principle:

$$u(a) = G(u_i(a_i) \mid i \in I)$$

Here the u_i are local utilities and G is some constructor function.

As an example, utilities in competitive markets are working hard to cut churn and win their customers' loyalty. Helpful, personalized services like high bill alerts and insightful call centers are giving them methods to succeed. Machine learning can help utilities take customer care a step further—helping them identify business at greatest risk of switching providers, listen to their concerns, and offer solutions.

8.3 Search

There is an old slogan saying that humans spend most of their time for sleeping and searching. Search needs a plan. In this respect search has something in common with prediction. This happens whenever one has more than one way to proceed. With many methods, if a search does not seem successful, backtracking to a branching point takes place. Search is based on a mechanism to identify a goal to be achieved. Search, in particular, plays a big role in machine learning.

8.3.1 General Search Model

We present here a formal model for search.

Definition

1. A search model consists of a tuple (S, T, G, C) . The set S is a set of states of the model, while $T \subseteq S \times S$ is a set of available transitions between states. For a set S and a set T , the pair (S, T) is called a basic *search model*.

A search instance is a pair (s, G) where $s \in S$ is the initial state and $G : S \rightarrow \{yes, no\}$ is the termination criterion, also known as the *goal test*.

Going from one state to another one along a transition is called a search step.

A control function C is a mapping generating the search steps. It selects the next steps: with $C(s) \in \{s' | (s, s') \in T\}$.

2. If a state with goal test value $\{yes\}$ is reached, then the search stops successfully.
3. A control function does not always exist. If it exists, the next search state depends only on the last state. The triple (S, T, C) is often called an autonomous search process.

A common example of search algorithm for this purpose is the gradient descent technique, where the neighboring state alternative with the smallest aggregated cost is selected next.

Now we assume that there is a definition of “best”. The search goal is to find best state sequence along a single path at certain time t , the best state and the best score.

Machine learning is a wide area. We just mentioned concepts of learning where some specific aspects are of interest for handling signal processes. The general purpose is to extract properties using given knowledge (mostly in the form of examples). A very popular type of machine learning problem is the unsupervised learning, where examples with unknown values for their hidden properties are classified. This is a procedure that naturally fits with stochastic processes. For general information on machine learning and Gaussian processes we recommend the popular book (Rasmussen et al. 2006). Search plays a major role in predictions. There one often has the choice between different alternatives and selecting the right one is a search problem.

8.3.2 *Preference Relations*

To understand utility one should look at its *relational version*. This is the preference relation that says which of two possibilities one should prefer and take. If one has several choices a preference relation brings them into a partial ordering. The top one is called the most useful one. The *functional version* is a real valued utility function where the higher values are the preferred ones. Utility theory is concerned with analyzing the relational and the functional versions.

Utility is not an absolute concept but can depend on the user. The user specifies preferences. However, typically the user does not have the preference ordering as a whole available, but rather only specific examples. What is useful to one agent may be rejected by another agent. In the view taken here, utility is a central concept of learning. This implies that what has been learned is used afterwards. The advantage is that it relates to an established theory namely the utility theory. It is also used in connection with similarity where the utility is studied in more detail. To determine what is the most useful learning, a search process is involved.

8.3.3 *Different Learning Methods*

In order to obtain an overview over the huge number of learning methods, research has introduced categories or types. Here follows a very general categorization that distinguishes three major types. They are supervised, unsupervised, and semi-supervised learning. In supervised learning, for each example one also knows the desired behavior expected from the learner, and the learning is itself controlled for this purpose. The unsupervised learning is central in this context because we deal with processes that have unknown desired response from the learner. That means that there is no teacher who helps and gives correctness or usefulness conditions for the single learning step. This includes the last learning step. There is nobody saying “stop, now you have it!”. Semi-supervised is something intermediate. For instance the teacher says that a real number must be positive but cannot predict the exact value. Later on we will come back with more details.

For each purpose there are some specific learning methods selected from the many available ones. For problems with variables that cannot be observed, unsupervised or some semi-supervised learning seems to be natural. One does not know the exact relations between data and desired behavior but one may know some properties of the data.

For the recognition problems of stochastic processes the main techniques discussed here are clustering and Bayesian reasoning. The latter helps with search.

Another at least as important aspect coming from the application of the properties is their usefulness. A good approximation may be as useful as a total truth. However, this aspect is also not clearly defined.

Usefulness is one of the key concepts in the book. There are many possible uses. The usefulness is known to also be subjective. That means, if something can be useful to one user or purpose while being uninteresting to another user or purpose. In this respect there should be someone who gives an order for the learning process steps. Often (mainly for unsupervised learning), the possible results of a single learning step are only approximately true. This is a statement of statistical character. In addition, the learning progress is difficult to determine.

There are different kinds of stochastic processes and therefore different kinds of successful learning methods.

Another aspect is the usefulness of the learned results even if they are uncertain. In Fig. 8.2 this is related to the change of the learner after learning. As mentioned, the usefulness of results often is subjective and can vary from user to user and situation to situation. Hence learning is not only a theoretical task (although it uses many theoretical methods and results) but ultimately a practical one. In Part III we discuss several practical applications that illustrate this.

8.3.4 *Similarities*

In order to compare several items, like stochastic processes, with respect to some property of interest (as for instance usefulness for a certain purpose) one needs a comparison concept. Often, the items in our context are methods. The concept should reflect the usefulness with respect to the purpose that helps to select the item. From a cognitive sciences perspective it is suggested to look at analogy. Analogy says, among others, that what was successful for one problem may be successful for another problem. Unfortunately analogy does not tell us how this goes technically. For this one needs another technical concept where the technical details can be checked.

Equivalently one can use distances. The first one says how close and the second one how distant two objects are. This is measured gradually. These concepts describe when and how one can transfer a problem solution by analogy. There are again very many similarity/distance concepts. They differ in how they define similarity in certain situations. Similarity will play a big role and we will encounter it several times. One special way is a very specific similarity concept formulated as distance and employed in the “[Dynamic Time Warping](#)” (DTW) method, frequently used for stochastic processes.

8.3.5 *Learning to Recognize*

Let us now consider the role of machine learning for recognition. Recognition means to discover certain properties of signal processes and this is what finally learning supports. Examples are the recognition of the meaning of words, or the existence of a disease, or the chance of an earthquake. It is easy to understand that this has something to do with prediction because it looks what presently happens in order to foresee the future.

The recognition concerned with stochastic processes has its own problems. An important technique consists in having special functions applied to the signals in order to handle them easier. Mostly, the example signals are far too many for handling them optimally in the presence of combinatorial explosion. During the effort to reduce this number, one has to take care and reuse the loss of important information.

An approach in this direction is to reduce each example by generating finite feature vectors, and the associated method is called feature extraction.

A basic question is: What is important and which information does one wants to keep? This is up to the user and the situation. Illustrative examples are found in the applications provided in Part [III](#). This question cannot be uniformly answered but depends on the specific application and the interest of the user. Here the difference between truth and usefulness is important. One can learn something that is true while being absolutely useless.

On the other hand, something that is not exactly true can still be very useful. As a consequence, learning depends on a situation that may change and also on user. A signal process may be dangerous in medicine but wanted in business. This indicates that learning alone is not sufficient. The usefulness also depends heavily on what one does after learning with the learned information. In Fig. 8.2 this is contained in the change of the learner.

8.3.6 Learning Again

Learning does not happen in an isolated way at a specific point of time. It builds upon and is shaped by what we already know. Therefore it is a dynamic process. This process has to be modelled. For learning properties of signal processes the learning process may take a long time or may never end. For humans there exists a concept of “lifelong learning”.

There are different kinds of features vectors and feature extraction solutions. In addition there exist processes of feature transformation. After some transformation a property that was hidden before may become obvious. Therefore feature transformations may be regarded as learning steps.

An important reasoning task that applies results of learning is prediction. A simple but typical example of signal processing is the prediction. It corresponds to questions like: Given a signal sequence S up to time n , what is the value of S at time $n+1$? Prediction is connected with many problems that have unknown solutions. Learning in support of prediction can be central in many applications, such as in business or medicine.

An application to the estimation of exchange rate is shown in Fig. 8.3. This process is stochastic with unknown probabilities. Figure 8.3 shows simulated exchange rates of stock value for some time interval in the past. The graph shows examples of exchange rate in blue, while in red it shows smoothed prediction

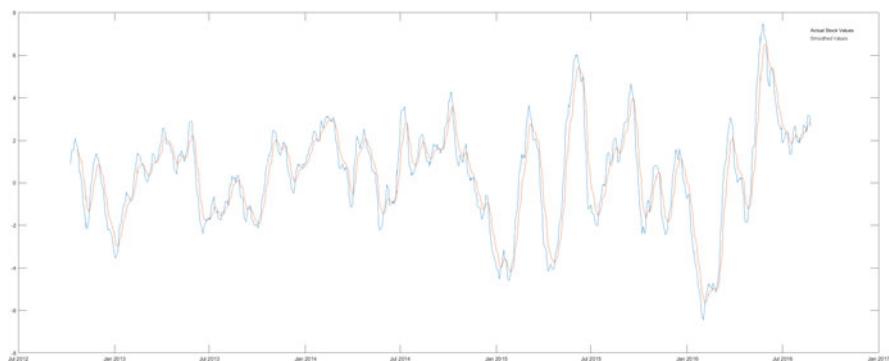


Fig. 8.3 Simulated stock value

rates. The two versions do certainly not fully agree. However, if one knows the predicted version in advance, its consideration will provide certainly utility when taking action.

8.4 Background Information

Traditionally, learning was an phenomena associated only with living beings. Humans learned by experience or by being told. Today one extends it also to machines by inspiring from how humans learned. This is distinct from the related area of machine supported learning for humans (Brown et al. 2014). The latter has to face the difficulty that human learning is to some degree hidden in the “black box” brain.

Learning can be seen as a consequence of many recorded examples. This will also be the view in this book. Examples help with acquiring new or modifying insights. It may involve different types of information. Most of the theories learned are not necessarily true. They are merely a guess that can, however, be very useful. The usefulness is a topic mentioned earlier.

General learning is discussed in Brown et al. (2014) and machine learning in Mitchell (1997). A modern view is found in Langley (2011). Brown et al. (2014) and machine learning in Mitchell (1997).

Learning is a complex process that may need a long time. A relevant special case is episodic learning. It changes a behaviors as a result of an event. A relevant method is canonical correlation analysis (CCA) which can be used to estimate local linear models.

Learning cannot be separated from the activities that are based on this learning. It changes the learner as shown in Fig. 8.2. After all, this is the overall purpose of learning. It can be performed by humans or machines. Humans can learn by themselves; where the learner and the receiver is the same agent.

An important issue is that a machine can learn to understand a command correctly for afterwards performing an action. This shows that learning and the following action are intimately connected (Intrilligator 1982). This is also related with forgetting learned principles if they are not satisfactory. Forgetting in learning is a special area in itself and is often important for a success. An early example of controlled learning is given in Miller et al. (1960).

Learning has many different interpretations. In this book we are concerned with automatic learning from observations and not with teaching in schools. This is addressed in the area of Machine Learning. A very popular and successful textbook is Mitchell (1997). The problem of blind signals deals with unknown sources of stationary processes with a zero mean. It is assumed that it is impossible to obtain the original sources. A specific method for learning such signal processes is given in Amari et al. (1996). For general learning one needs to have observations. Obtaining useful observations is sometimes difficult and it is an essential prerequisite for successful learning. The number of observations may be very large and for a human

they may be difficult to process. This applies also to signal processes and it is a major reason to use machines. Later on, we will return to specific methods and applications with respect to signal processes. Learning is connected with signal processes because often only certain probabilities are available, if any.

The fact whether a machine learns the goal should be precisely defined, so that one can detect or measure the success. This would allow for comparing different learning methods. The issue has mainly been studied in connection to applications or neural networks. In the latter, machine learning methods are essential. For utility theory see Brown et al. (2014), Barten and Boehm (1982).

8.5 Exercises

Exercise 8.1 Give an example where learning allows for solving more problems. Show a learning method for this.

Exercise 8.2 Give an example for learning that is useless if it is not followed by proper subsequent actions.

Exercise 8.3 Give an example of a learning problem for signal processes. Look at the applications given in Part III.

Exercise 8.4 Give a learning example where forgetting plays a crucial role.

References

- [Brownetal2014] Brown, Peter C., Roediger, Henry L., McDaniel, Mark A. (2014). *Make It Stick: The Science of Successful Learning*. Cambridge, MA: Belknap Press.
- [Mitchell1997] Mitchell, T. (1997). *Machine Learning*. McGraw Hill, 1997.
- [Langley2011] Langley, P. The changing science of machine learning. *Machine Learning* , 2011
- [Miller1960] Miller, G.A., Galanter, E., Pribram, K.H. (1960). *Plans and the Structure of Behavior*. Holt, Rinehart & Winston, New York.
- [Amari1996] Amari, S., Cichocki, A, Yang, HH. A new learning algorithm for blind signal separation, 1996.
- [BartenandBoehm1982] Barten, A., Boehm, V. (1982). Consumer Theory. In: Kenneth J. Arrow and Michael D., 1982
- [Intrilligator1982] Intrilligator (eds.): *Handbook of Mathematical Economics*. Vol. 2. North Holland, Amsterdam, 1982
- [Rasmussen2006] Rasmussen, C. E., Williams, C. K. I., *Gaussian Processes for Machine Learning*, the MIT Press, 2006, ISBN 026218253X.

Chapter 9

Signal Processes, Learning, and Recognition



Overview

This chapter discusses topics of probabilities with interest to digital processes and their applications. Some emphasis is put on stochastic processes, presented in detail later in Chap. 13, namely Markov processes, Hidden Stochastic Model Processes (HSM) and Gaussian Mixture Models (GMMs). In HSM the states of certain relevant processes that influence the signal are not directly measurable and the GMMs contain a mix of different Gaussian probabilities. Much interest is put on recognition and search, in particular Bayesian reasoning. Different learning processes including some generalities for signal processes are presented in this chapter.

9.1 Learning

Learning is a process. In the learning process one formulates how the available data is organized in order to conduce to the best learning outcome. The outcome is approximated step by step and there may be no termination of the learning process. In fact, there may even not be a precise description of the learning goal. Typical examples are frequently found in biosciences. Figure 9.1 shows the typical learning process in machine learning.

Next we consider the task in more detail and as a whole. The observed variables of the signals have directly accessible values from their domain; they correspond to inputs or to problem attributes. The case of supervised learning is shown in Fig. 9.2:

In general, the task is to get some additional information about the situation using signal processing. Tasks can be of quite diverse nature. Some examples of questions and (possible) answers are:

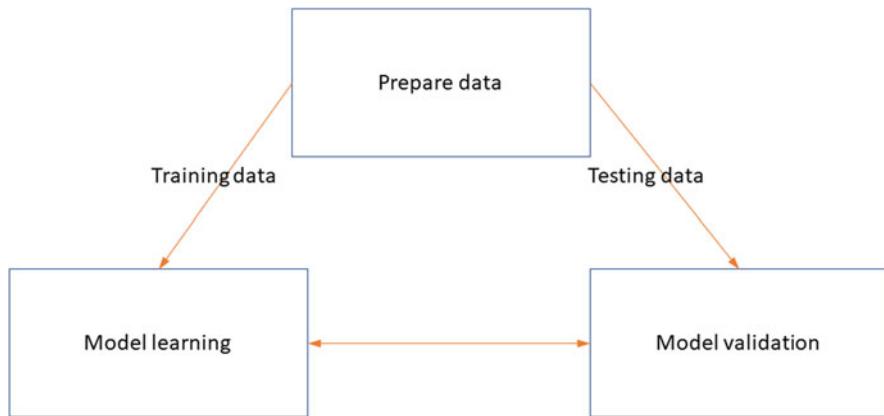


Fig. 9.1 Typical machine learning process

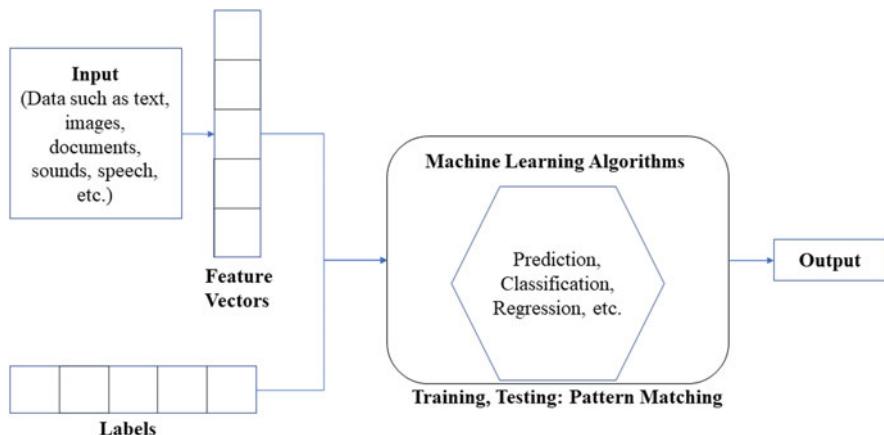


Fig. 9.2 General supervised learning

1. Question: What is the pulse frequency of this patient right now?
Answer: Take a measurement (only signals are needed and no learning).
2. Question: What is the capital of India?
Answer: This case is supervised learning. The teacher is the map - look at the map.
3. Question: Given a set of family members. For every pair of persons one sees the family relation. What is the whole family structure?
Answer: The result reflects the truth. Take a supervised learning algorithm that knows the family structure.
4. Question: Suppose you have a medication against blood pressure. What is better, to take it each day or every second day?
Possible answer: Make a series of tests and compare them using DTW (A signal process is needed).

5. Question: What is the danger that persons with a blood pressure over 180 will die within the next 8 month?

Possible answer: Collect statistics from a hospital. Arrange the results in clusters “die” and “not die”. Derive the result from the comparison of the clusters.

To answer these questions one has different kinds of learning situations and therefore different learning methods, too. The learning topics cover a huge and diverse area. This text is restricted to situations related to signals and signal processes. An important learning technique is the widely used classification technique called Bayesian method. We started the description in Chap. 8. We further extend the discussion by providing more details. Bayesian reasoning contributes probabilistic knowledge. It is concerned with the influence relations. An influence diagram is represented as a graph where the nodes are properties or decisions, and edges show conditional independence relations. An edge from a node A to a node B indicates that A has a direct influence on B . However, from such a diagram one cannot see how large the influence is. Influence can be coded by a weight for quantifying the influence. In a probabilistic setting the influence is represented by conditional probabilities. Bayesian networks use this information and can be applied to answer probabilistic queries about unobserved variables.

1. In supervised learning the result has to be precisely determined and specified. In supervised training procedures one uses only examples with known desired response, for example, correct or incorrect. The terms correct and incorrect are used as true and false in classical logic. For learning a concept description, a comment is added to each example specifying whether it is correct or not. The agent who provides the comment is called the teacher. An example of supervised learning algorithms is offered by back propagation for neural networks. This version of learning requires at least that the concept to be learned is of a precise character and no uncertainty is involved. An example is when one would guess the next letter in a long word that at least is known to the teacher. For signals it is only applicable if there is a specific rule generating the classifying comment. For our purposes supervised learning plays a small role for stochastic processes and we will not otherwise discuss it.
2. In unsupervised learning there is no such teacher. This is the case for HSM. However, there is sometimes a subject that plays partially the role of a teacher. It is assumed that teacher may (or even should) have some general knowledge about the domain. This leads to a clever way of handling the problem. An example is application of a similarity measure. The trained data are used to get some pattern and these patterns are used as identifications of the unknown data. For the results the term correct does not apply; one gets only guesses. Sometimes there are no correct answer. For example: What is the best heart rate? One may know what an ideal heart rate is but “best” heart rate requires further investigation.
3. The semi-supervised learning falls in between the supervised and unsupervised learning; some partial correctness comment is given. Semi-supervised can for instance mean that one knows that the result is a real positive number.

9.2 Bayesian Formalism

The main contribution of Bayes, is in so called Bayesian reasoning. It quantifies the influence in terms of probabilities. The way to apply Bayes rule is first by annotating classes that are used in examples at hand.

Definition A Bayesian network is a directed graph with the properties.

1. The nodes are random variables and the labels on the nodes are conditional probability tables between the probabilities of the two nodes are attached.

The general situation is now as follows. Suppose an event E and a number of hypotheses H_1, \dots, H_m are given such that

- Each hypothesis has a known a priori probability $Prob(H_i)$
- For each H_i the conditional probability $Prob(E, H_i)$ is known.

To decide a choice among the chosen hypotheses one takes the one where the a posteriori probability $Prob(H_i|E)$ after the event E has happened is maximal (maximum likelihood principle). The resulting type of network is denoted BBN in short for Bayesian Belief Network.

9.2.1 Dynamic Bayesian Theory

Networks discussed so far have been of static character. The dynamic events come in when a new event takes place that gives a specific value to some variable that was unknown before. Then other probabilities will change and this propagates through the network. This happens frequently when planning and executions are interleaved. In the belief network this means some node that is not an initial node gets as an observed value rather than only a probability distribution. As a consequence, the probabilities in the network have to be recalculated when such new events come in.

9.2.2 Recognition and Search

Suppose an event E has happened and a number of hypothesis H_1, \dots, H_m for events is given such that:

- Each hypothesis has a known a priori probability $Prob(H_i)$.
- For each H_i the conditional probability $Prob(E|H_i)$ is also known.

The goal is to determine now the hypothesis best explaining E . To decide on the choice among hypotheses one maximizes the a posteriori probability $Prob(H_i|E)$ after the event E has happened. Unfortunately, this is not given directly and one has to find out this probability. Figure 9.3 helps in visualizing the application

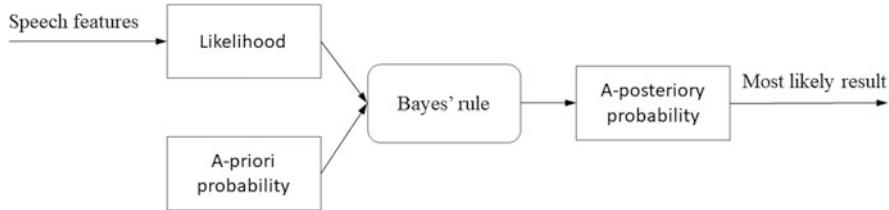


Fig. 9.3 Bayes rule for computing probabilities

of maximum likelihood principle. This counts under Bayesian reasoning. The Bayesian Belief Network (BBN) is introduced in Chap. 10. The wanted conditional probability of the H_i given E can be computed by Bayes' rule:

$$Prob(H_i|E) = \frac{Prob(H_i)Prob(E|H_i)}{\sum_{j=1}^n Prob(H_j)Prob(E|H_j)} \quad (9.1)$$

Bayes' rule allows transforming the a priori probability of the hypothesis into some a posteriori probability. In applications, the former is usually unknown while the latter is known or at least can be computed by repeating the cause and accounting the observations. One sees also that probabilities can change when an event has occurred (an event observed to occur has probability 1). An immediate consequence is that one gets a new network. This is obtained by replacing a node with a label H_i by the label $Prob(H_i|E)$. This is illustrated in Fig. 9.3 depicted below. Here the network should not be static. New events happen that could change the network as indicated. Therefore one considers now dynamic networks in which labels can be changed. Dynamic networks appear quite often in reality. They are at the foundation for all kinds of Bayesian reasoning concerning dynamic phenomena. This was one reason for the name “Belief networks”: More details about this concept are introduced in Chap. 11, specifically in Sect. 11.3.3.

Example in a Forward Mode

Suppose we have the edge A and B , then we get

$$P(B) = P(B|A)P(A) + P(B|\text{not } A)P(\text{not } A)$$

and

$$P(\text{not } B) = P(\text{not } B|A)P(A) + P(\text{not } B|\text{not } A)P(\text{not } A)$$

Example for Modeling

Suppose we have some commands in the vocabulary list: “Stop”, “OeffnedasFenster”, “Gehweiter”. Suppose further that we have a 3 states model for the command “OeffnedasFenster”. They are:

1. oeffne
2. das
3. fenster

and the three states are denoted by q_1, q_2, q_3 . We have one state model denoted by q_1 for the command “Stop” and two states model q_1, q_2 for the command “Gehweiter” and “Geh” “Weiter” Important additional concepts are listed below:

- Acoustic features are obtained by the APLTT feature extraction. Each feature vector has several feature elements. The APLTT feature vectors for each command are denoted by o_1, o_2, \dots, o_T .

These feature vectors are used in the Gaussian mixture model to obtain the probability density function (pdf). Given the features and the model, i.e. the pdf of the acoustic speech features, we compute the likelihoods of the states given the features and the model.

Likelihood

This gives us the probability of possible features given all possible states. This means for example $p(o|Oeffne\ das\ Fenster_{q_1, q_2, q_3})$ p denotes the probability measurement.

- Utilizing Bayes rule we can find the most likely answer, i.e., what is the probability that states belong to the given features and the model that is $P(\text{oeffne das Fenster}|o_1, o_2, o_3, \dots, o_T, \lambda)$.
- A-priori probability is the computation of the pdf by the Gaussian Mixture Model (GMM).
- A-posteriori probability indicates the likelihood of the state given the features and the model.
- The highest probability indicates the answer to the problem.

In the next section we show the Hidden Markov Model (HMM) architecture and the computational methods to obtain the HMM parameters. We discuss this first informally.

9.2.3 Influences

Up to now we have used the term influence several times without giving a definition because it is an everyday term that has an intuitive meaning. We referred mainly to this intuitive meaning of the concept. Now a formal definition is introduced.

Definition Suppose A and B are two events. Suppose there is a set of admissible changes. A has an influence on B if some changes of A result in changes of B . One should observe that an event can have several influence factors. This reflects the intuitive meaning of influence. There are many concepts to describe influences computationally and formally. In probabilistic terms influence for instance can

be expressed as a conditional probability. This makes an influence visible and quantifiable. However, in most situations influence is hidden. This motivates machine learning tasks trying to determine the influences. Weights can be used to represent influence. A natural task is to use a Network for finding the values for the weights in a linear weighted measure. The weights influence the importance of the attributes and this can be influenced by different factors that may accumulate in a complex way. This shows also that an influence can come from different sources and one has influences of different strengths. In a probabilistic model one can regard each attribute as a random variable with values from its domain. One can assume that the data vectors are distributed according to some probability distribution.

9.3 Subjectivity

In the mathematical definition of probability, for instance following the Kolmogorov approach, subjectivity plays no role. It is purely based on frequency. However, when people talk to each other often they refer to probabilities without having any frequencies. That means they have estimates without giving a precise reason for them (that is acceptable for everybody). This also plays a role when humans judge about signal processes. Subjective probabilities are quite common among humans. A problem is how to handle them. There are some questions connected with it:

1. What does subjective probability tell us?
2. In how far should we believe in them and can we improve them?
3. What kind of actions should be based on them?
4. Are the actions the same as if the probabilities are based on frequencies?

Some of the answers to the posed questions are presented below: The answer to the first question is that the person who tells it would almost behave as if the statement would be true. Hence it is a useful information for the listener. The second question refers to the reliability of the speaker. A support would be if one knows something from the subject and about the speaker that is helpful. The answer to the third question uses the same assumption under which the second question is answered: Behave in the same way unless one gets a reward from the subject. The answer to the fourth question would say that the actions would be almost the same. One would only be reluctant for actions that are risky.

9.4 Background Information

General Discussion and Past Work

Basic work on probability was the work of Kolmogorov. A first book is Kolmogorov (1933). Stochastic processes were first studied in the late nineteenth century. The first person to describe the mathematics behind Brownian motion was Thorvald

Thiele in a paper on the method of least squares published in 1880, Thiele (1880). There is a classic reprint of that original work. This was followed by Louis Bachelier in 1900 in his PhD thesis “The theory of speculation” (Bachelier 1900). Albert Einstein (in one of his 1901 papers, Einstein (1901)) brought the solution of the problem to physics. Networks are discussed in Pearl (1988) and Jensen (2001). Probability occurs in mathematics as well as in every day conversations. It is generally used for describing a great deal of uncertainty. The term is roughly synonymous with plausibility. It has reference to reasonableness of belief or expectation. There are many ways to give a basic formulation. Today that frequently used term goes back to Kolmogorov. This requires quite an amount of knowledge concerning frequencies. However, one should also know that in every day communication it is used informally and in a subjective way. For instance, if the fever is high what is the probability for high fever next week? The answer depends on who posed that question. See Fishburn (1986) further details on this question. Subjective probabilities are defined in terms of a person’s preferences, in so far as these preferences satisfy certain consistency assumptions. A basic contribution to statistic including subjective aspects is in Savage (1954). This describes also modelling probabilities. A relation to this utility, it is discussed in Davidson, D. and Suppes, 422–443, (1956). For signal processes see Shynk (2012). Gardiner (2004), provides inside to natural sciences.

Bayesian Probability was named after the English mathematician Thomas Bayes. Bayesian decision theory came long before Decision Tree Learning and Neural Networks. It was studied in the field of Statistical Theory and more specifically, in the field of Pattern Recognition Jensen (2006). Bayesian Decision Theory is the basis of important learning schemes such as the Naïve Bayes Classifier, Learning Bayesian Belief Networks and the EM Algorithm. Bayesian nets take care of the fact that often new knowledge in the form of new events coming in, see Rasmussen (2006).

Suggested Reading

Jensen (1996), provides a complete introduction to Bayesian networks. Another special learning method is Q-learning Watkins (1989), <https://weka.8497.n7.nabble.com/New-release-of-YALE-3-4>. It is a simple way for agents to learn how to act optimally in controlled Markovian domains. For learning tools in general Yale 3.4 and Weka are recommended.

For more information, readers are encouraged to read the used references in this chapter such as [Allen2004], [Kaelbling1998], [Sonnenberg1993], [Lanchantin2005], [Satish2003], [Myers1981], [Chen2009], [Rabiner1993], [Richter2013], [Brinksma1995].

9.5 Exercises

Exercise 1 Make the reasoning in the above questions (3) and (4) precise.

Exercise 2 Store the conditional probabilities of 10 two-valued variables as a table. How much storage space do you need? How much can you save using the Bayesian network representation?

Exercise 3 Find areas in signal processing where subjective probabilities dominate and give reasons for that.

Exercise 4 Give an example about signals where almost all probabilities are subjective.

Exercise 5 Find a Bayesian Network that describes a football game.

Exercise 6 Find areas in signal processing where subjective probabilities dominate and give reasons for that.

Exercise 7 Give an example about signals where almost all probabilities are subjective.

Exercise 8 Find a Bayesian Network that describes a football game.

References

- [Rasmussen2006] Rasmussen, C.E. (2006): Gaussian processes for machine learning. MIT Press.
- [Jensen1996] Jensen, F.V. (1996): An Introduction to Bayesian Networks.
- [Jensen2006] Finn V. Jensen, F. V. and Nielsen, T. D., Bayesian Networks and Decision Graphs, Springer, February, 2006
- [Watkins1992] Watkins, J.C.H., Dayan, P. (1992): Q-Learning. Machine Learning 8, p. 279–292, Kluwer Publ. Co., 1992
- [Kolmogorov1933] Kolmogorov, A.N. Grundbegriffe der Wahrscheinlichkeitstheorie, Berlin, Springer, 1933
- [Thiele1880] Thiele, Th. Theory of Observations (Classic Reprint), 1880
- [Bachelier1900] Bachelier, L., Annales scientifiques de l’École Normale Supérieure, Serie 3, Volume 17 (1900), p. 21–86, 1900
- [Pearl1988] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kauffmann Publishers, San Francisco, 1988
- [Einstein1901] Einstein, A. (1901) Conclusions Drawn from the Phenomena of Capillarity (Annalen der Physik 4.)
- [Jensen2001] Jensen, F.V., Bayesian Networks and Decision Graphs. Springer-Verlag, New York, 2001
- [Savage2014] Savage, L.J., The Foundations of Statistics. New York, Wiley, 1954
- [Fishburn1986] Fishburn, P.C. (1986): The Axioms of Subjective Probability. Statistical Science 1, 1986
- [Shynk2012] Shynk, J. (2012). Probability, Random Variables, and Random Processes: Theory and Signal Processing Applications, Kindle Edition, 2012

- [Gardiner2004]** Gardiner, C., *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*, 3rd ed., Springer, 2004.
- [Davidson1956]** Davidson, D. and Suppes, 422–443, (1956), A finitistic axiomatization of subjective probability and utility. *Econometrica* 24, 1959
- [Watkins1989]** Watkins, C. *Learning From Delayed Rewards*, PhD thesis, Cambridge, 1989

Chapter 10

Stochastic Processes



Overview

A stochastic process is a mathematical construct that usually is defined by a group of random variables. Examples that are governed by stochastic processes include the growth of a bacterial population, an electrical current fluctuating due to thermal noise, or the movement of a gas molecule. Stochastic processes have applications in many disciplines such as signal processing, speech and image processing, computer science, just to name a few. Specifically, the theory of stochastic processes addresses a collection of random variables that are indexed by some mathematical set. The random variables have the collection of the values taken from the same mathematical space, known as the state space. This state-space could be represented by a set of integers, of real numbers, or Euclidean space of arbitrary dimensions. In applications many signals occur that are of complex nature and cannot be predicted. These signals have a source that is known only roughly. The information about them is not isolated but rather stems from a whole set of signals. The signals are thus separated in groups called signal processes. We assume that the signals in a signal process occur according to a partial ordering, like for instance in a sequential process. This is motivated by the fact that signals are created by events that occur in time.

The process has additional properties like time/space distances between the signals or constants of varying strength. Such processes have been studied in various ways. Examples are given in the applications in Part III of the book. One is interested in processes as a whole and their properties, not only in the individual signals.

Sometimes one can perceive the signal processes in a way that is regulated by logical and deterministic rules. Often this happens for artificially made signals generated by machines. It requires some knowledge about what is happening. When the rules are present and known one derives the properties of the process by logical reasoning using those rules. Unfortunately this is not always possible, as there may be no such rules or at least, one does not know the rules. This is mostly the case when the signals are generated by a biological source. The next simplest possibility

is that the process is organized according to probabilities that can be estimated. Then one calls it a stochastic process. An example of special and very common stochastic processes are the Markov Models. If the involved process states cannot be observed and measured directly, one calls them Hidden Stochastic Models (HSM). They are very common too and we concentrate on them for modeling purposes.

10.1 Preliminaries on Probabilities

Probabilities provide a means for analyzing stochastic processes. We will not introduce rigorously the subject but assume some familiarity with it, and occasionally clarify details and terminology. We denote probabilities by either P or $Prob$. Conditional probabilities of A given B are denoted by $P(A|B)$. We should remark that the assumption is about the event B and not about some kind of probability of B . As said, usually the signals do not occur isolated but there are rather very many of them, collected in a set.

Mostly, these sets are structured by a partial ordering. An example of the ordering principle is time, describing when signals occur, i.e. sooner or later or at the same time. For simplification it often suffices to restrict to linear orderings to clarify the principal concepts and processes.

Mostly the processes occur in dynamic problems, namely where time plays a role. There are different kinds of stochastic processes. The methods are specific for the types. In this context one calls the occurrence of the individual signals, event. In general, an event in such a process with time order is dependent on previous events. The dependency may be deterministic or not. Often the character of the dependency is described statistically.

10.2 Basic Concepts of Stochastic Processes

Intuitively even deterministic processes may look stochastic because their structure is unknown and complex. The general assumption is as mentioned that events in a process depend on previous events. This dependency is of at most probabilistic character.

One distinguishes between stationary and non-stationary processes. The first ones have the same statistical characteristics regardless of the time. That means the parameters like the mean and the variance of the probability model of the process is time-invariant. In our intended applications we deal mostly with stationary processes.

Now we consider some major types of stochastic processes. A basic concept is a filter as discussed in Part I. There are several kinds of filters. Two important ones are presented in Figs. 10.1b, and a.

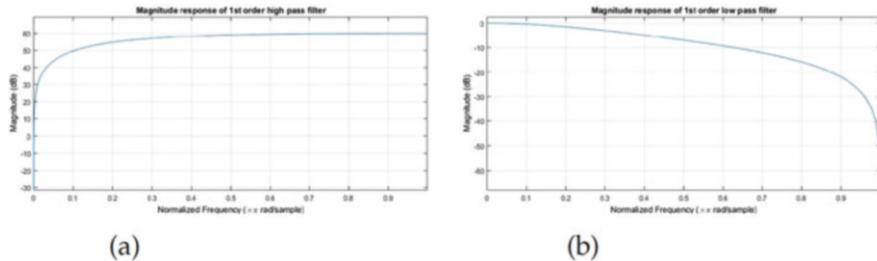


Fig. 10.1 The figure shows 1st order magnitude response of low-pass and high-pass filters. **(a) High-Pass Filter.** **(b) Low-Pass Filter**

10.2.1 Markov Processes

A Markov process is a stationary one where the state depends on time and the probability of each event characteristic depends on previous events. It has an additional dependence assumption. For example, first order Markov processes require that the outcome at time t is independent of all events prior to $t - 1$, given the event at time $t - 1$. The dependency is only on events (and not on probabilities) at time $t - 1$. If the processes events at each time t are described by random variable X_t with actual value s_t then the Markov condition in terms of conditional probabilities is:

$$Pob(X_t = s_j | X_{t-1}, X_{t-2}, \dots, X_0) = Prob(X_{t-1} = s_{j-1}) \quad (10.1)$$

That implies that a Markov Process has almost no memory. The basic assumption is that a prognosis made on the knowledge of a very short history is as good as made on a longer history.

Higher order dependency assumptions assumes the dependence on several previous events. In the next section we will generalize the Markov property somewhat in this direction. One distinguishes between discrete and continuous processes. A discrete process is also called Markov Chain. Equation 10.1 is based on the common notation for Markov chains.

There is no restriction on the possible probability distribution function in a Markov process. It is very common to use a Gaussian distribution or one of its generalizations as Gaussian Mixture Model (GMM) (introduced later). In any case, the probability function must be known for making further decisions.

If the set of possible values of the states is finite then one can describe the Markov Process by a matrix called the transition matrix.

Figure 10.2 shows a Markov chain with three nodes. The connections go from left to right, the probabilities are not shown.

The nodes shown with circles in the Markov chain illustrate states, typically associated with random variables. An arrow in a Markov chain shows that the state

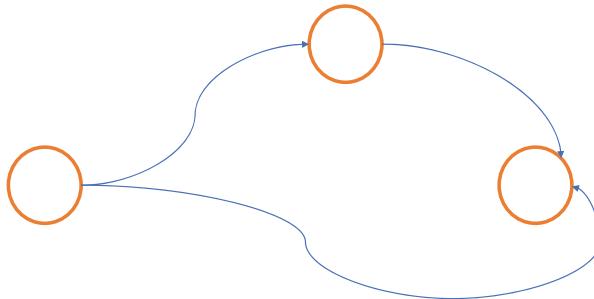


Fig. 10.2 Example of a Markov chain

pointed by the arrow has a conditional dependency on the value of the state from which the arrow leaves.

10.2.2 Hidden Stochastic Models (HSM)

Often, the state of a signal is at most partially or approximately observable. Such situations are considered in the area of **Hidden Stochastic Model** (HSM). Originally this was (as the original name HMM indicates) an extension of the Markov Model. Here we present a more general view. We use many concepts and technical details of extensions of those that have been developed for Hidden Markov Models.

HSMs are used for modelling general stochastic processes with some unobservable random variables. A general description of the situation follows. One starts with a set $O = \{o_1, o_2, \dots, o_T\}$ called observations. The elements of O are called events because they can be seen and are therefore known to the user. In a stochastic situation they are a basis for predicting future events. Further there are N states $Q = (q_1, \dots, q_N)$ of the system. This is another basis.

The states are not seen and unknown to the user in an actual situation. However, one may know certain events related by known probabilistic relations to these states. In a state q_k one rather knows the probability distribution $p_k(o_i)$ of the output for the observed event o_i .

There are two types of statistics involved and assumed to be known in an HSM:

1. The first is a transition probability between states:

$$A = \{a_{i,j} \mid 1 \leq i, j \leq N\} \text{ with } a_{i,j} = \{q_{t+1} = s_j \mid q_t = s_i\}$$

This describes the relation between two states.
2. The second is the event output probability in a state state:

$$B = \{b_k(j) \mid 1 \leq j \leq N, 1 \leq k \leq M\}$$

In general, each event takes place in some state. Initially we have a starting state called π . Notation: For this we have different choices ($\pi = \pi_i \mid 1 \leq i \leq N$) with

$\pi_i = \text{Prob}(s_1 = q_i)$. That means every state can be the starting state. This leads to the following definition.

Definition A stochastic model is of the form $\lambda = (A, B, \pi)$. One should notice that this model provides implicitly the numbers N and M . Again we have here the distinction between stationary and non-stationary processes.

Suppose now that in some situation there is a sequence $\mathbf{O} = (o_{i1}, \dots, o_{in})$ of observations of signals. Given a stochastic model $\lambda = (A, B, \pi)$ one may ask now what the probability of such an observation sequence and a given state sequence q given this model is? That means one wants to estimate:

$$P(O, q | \lambda) = P(O | q, \lambda)P(q | \lambda) \quad (10.2)$$

This probability is of interest and can be a learning goal. Next we deal with some general properties and methods of stochastic processes and problems associated with them. They play a role for the analysis of GMM too. One will always start with an initial estimate of the model parameters. Then one describes the iteration and finally the termination. For the learning one needs additional concepts that we will consider now. Below we will discuss situations with several processes. This has an impact in so far as many probabilities occur simultaneously. We introduce some standard concepts. The joint cumulative distribution function F of two distributions is given by $F(x, y) = P(X \leq x, Y \leq y)$. From this we get:

$$P(X \leq x) = F(x, \infty)$$

$$P(Y \leq y) = F(\infty, y)$$

$$P(X = x, Y = y) = P(x, y)$$

If both X and Y are continuous, then a joint probability density function (pdf) f will generally be:

$$F(x, y) = \int_{-\infty}^x \int_{-\infty}^y f(x', y') dx' dy'$$

Definition A collection of X_1, X_2, \dots, X_n is mutually independent if

$$P(X_1 \leq x_1, \dots, X_n \leq x_n) = \prod_{i=1}^n P(X_i \leq x_i)$$

The co-variance between X and Y is defined by:

$$\text{cov}(X, Y) = [(X - E(X))(Y - E(Y))]$$

10.2.3 HSM Topology

The topology of a directed network is the ordering in which nodes are connected. The HSM topology gives further information about the model. It says what type of a topological HSM model one selects for a particular pattern recognition problem. It must be defined in advance, for example:

- If the HSM is discrete or continuous;
- If the model is left-to-right or fully-connected;
- If the transition probabilities are fully defined such as in the transition probability matrix.
- As indicated earlier, the model may be stationary (i.e. fixed over time) or variable.

We consider discrete and continuous situations. Our topology further described in Part III is left-to-right but that is not a real restriction. Next we discuss some methods for working with a model and learning its properties and methods. The methods are used to simplify the situation without losing essential information.

10.2.4 Learning Probabilities

In hidden stochastic models the value of certain states are not observable. On the other hand they are needed for certain computations. A suitable way to get them, at least approximately, is to apply a learning method to the provided events. A standard way both to solve the probability problem and to take prior knowledge into account is to use a pseudo count or prior count for each value to which the training data is added. A pseudo count is an amount added to the number of observed events in order to change the expected probability, for example, avoiding zero probabilities just because rare data does not occur in training.

Suppose there is a binary feature Y , and an agent has observed n_0 events where $Y = 0$ and n_1 events where $Y = 1$. The agent can use a pseudo count $c_0 \geq 0$ for $Y = 0$ and a pseudo count $c_1 \geq 0$ for $Y = 1$ and estimate the probability as:

$$P(Y = 1) = \frac{n_1 + c_1}{n_0 + c_0 + n_1 + c_1}$$

The simplest approach is to add one to each observed number of events including the zero-count possibilities.

10.2.5 Re-estimation

A variable model has to be adjusted from time to time. The re-estimation method adjusts the model for maximizing the probability $p(O|\lambda)$ of the observation vectors O . The re-estimation adjusts the parameters. The point is that one wants to change the parameters such that result is really an improvement in the sense of the intended utility. The following questions have to be answered in order to solve the problem of re-estimation.

- How do we improve our estimates of the HSM parameters?
- How do we define or measure the term “improve” of the parameters?

The first question is attacked by recursion. The second one uses the Noise to Signals measurement (SNR) and recursion. These questions will be discussed next.

10.2.6 Redundancy

Data or intervals are redundant if they are unnecessary or even disturbing with respect to the intended purpose. They can be removed whenever they are detected. However, this is not so easy to find out. A special case is silence where no signals come in. This will be discussed later. The intervals of redundancy are characterized by thresholds and those should be discovered. However, one should handle redundancy with care. Often, it is not a purely syntactic property but a semantic one too. Silence can provide an important information; this is discussed below. The redundancy of the collected data is handled in removing and pre-emphasizing by the following steps. A measure of redundancy between two variables is the mutual information or a normalized variant. A measure of redundancy among many variables is given by the total correlation.

Finally we come to probability. Each duplicate component added to a system decreases the probability of system failure according to the formula.

$$p = \prod_{i=1}^n p_i$$

where

- n = number of components
- p_i = probability of component i failing
- p = the probability of all components failing (system failure)

Here we assume that the failure events are independent.

10.2.7 Data Preparation

The redundancy of the collected data is handled in the first place by ignoring it or by removing and pre-emphasizing by performing decimation and aliasing procedure. This is a process that reduces the sampling rate by a factor. The decimation factor is usually an integer or a rational fraction greater than one. This factor multiplies the sampling time or, equivalently, divides the sampling rate. One way to do this would simply be down sampling. Down sampling is the reduction of the process by omitting certain samples. The problem is that it causes the high-frequency signal components to be misinterpreted by subsequent users of the data. One calls this aliasing. The first step, if necessary, is to suppress aliasing to an acceptable level. The filter designed to perform this is called an anti-aliasing filter. For the decimation, one should first down sample as long as no information is lost. For instance, if the signal is band-limited from 200 to 3500 Hz one can down sample mostly to 16,000 Hz. Therefore one has the steps:

1. Down-sampling
2. Anti-aliasing filter

For anti-aliasing filter one can use a low-filter.

Example We consider the simple detection of aliasing in stationary signal processes. We take a process with randomly shifted periodic signals. We have a waveform (t) , with period T and we can produce a stationary process by adding to t a random time shift, l , which is evenly distributed on $[0, T]$. A sample path of our process then has the form $x(t + l)$. Then we have

1. $x(t) = \sin(2\pi ft + 2\pi fl)$ where l is evenly distributed on $[0, f - 1]$. If we under-sample with a sampling interval Δt , corresponding to the Nyquist band $[-(2\Delta t) - 1, +(2\Delta t) - 1]$, and then we reconstruct via convolution with the *sinc* filter and get the sine process given by
2. $xr(t) = \sin(2\pi \hat{f}t + 2\pi fl)$. Here, \hat{f} is the aliased frequency, given by $\hat{f} = f + k\frac{f}{\Delta t}$ where k is the unique integer that places \hat{f} in the Nyquist band. The key point is that the phase of the reconstructed signal is the same as the phase of the source even though the frequency has changed to the aliased value \hat{f} . For a process with a single harmonic, the reconstructed signal remains stationary because the phase term, $2\pi f\theta$, is evenly distributed on 2π .

Next we show a diagram explaining the frequency using a spoken phrase.

10.2.8 Proper Redundancy Removal

This procedure has several sub steps. The sub-steps are:

1. Apply decimation
2. Compute the signal envelope
3. Select the thresholds.

The thresholds are the start and end points of the intervals. The thresholds define intervals, in particular intervals that can be removed. The first step, decimation, has already been introduced. The signal envelope and the threshold will be introduced next. All three steps reduce the redundancy and shorten the signal. It has a computational benefit due to the fewer samples in the signal processing but can also loose important information. One can also omit time periods where nothing happens, i.e. no signals are present. These intervals are called silence, but see the problems connected with it. Here we include a warning: Often silence is not neutral. It may contain important information; see Part III. For instance, if the heartbeat is silent this indicates an important message. Also, in speech a silence can be an intended message. In the section on psycho-acoustic phenomena in Part III such cases are discussed.

10.3 Envelope Detection

The purpose of an envelope is to locate the signals properly. The concept of an envelope of a varying signal comes from geometry. It is a smooth curve outlining its extremes. It provides a boundary which contains the signal. The envelope of a signal is also an estimate of the signal level. The envelope procedure takes a signal as input and provides an output which is the envelope of the original signal. A problem is that the envelope should not be unnecessarily large. In fact, it should be as small as possible.

Computing the envelope is not trivial. One way of computing the envelope of the signal is by the Hilbert transform of the signal. The basic goal of the Hilbert transform in the time domain of a signal is to get another time domain of the signal that is helpful for determining the envelope. The Hilbert transform shifts the frequency components of the signal by 90 degrees but it does not change the amplitude. The Hilbert transform acts as a differentiator to a constant signal. This means if the signal has any constant component, the Hilbert transformation of the signal cancels it. This is equivalent to getting the zero mean of the signal. The signal is processed under the assumption that it is an ergodic process. In this process the time average of the signal is equivalent to the ensemble average of the signal. The importance of this is that the time average of the signal can be computed easily but the ensemble average cannot. The Hilbert transform of the signal $s[n]$ is denoted by $s_H[n]$. A formula for the computation is shown in Eq. 10.3.

Definition Hilbert transform:

$$s_H[n] = \frac{1}{\pi n} \otimes s[n] \quad (10.3)$$

The symbol \otimes denotes convolution. In Eq. 10.3 we see the computation of the envelope and the Hilbert transformed signal $s_H[n]$ using the real valued signal $s[n]$.

$$|s_H[n]| = \sqrt{s[n]^2 + s_H[n]^2} \quad (10.4)$$

The computation of the envelope using the Hilbert transform also provides a representation of the signal. The envelope of the signal using the Hilbert transform sets a qualitative boundary around the silence too. We have used this computation to obtain the envelope of the signal. Then we have selected the threshold. We exclude all the data that fall below the threshold and remove pause, silence and other redundant data. The advantage of the smoothing is that we get less computational costs. Provided Fig. 10.3 below shows a FFT of a frame as an example of spoken speech ‘Oeffne die Tuer’, the envelop detection by the LPF method and Hilbert transform. In LPF method, the signal is filtered using Butterworth worth filter, then low pass filtered prior to take the energy and compute the spectrum.

In Fig. 10.4 first we see the speech spectrum (see Part III) of “Oeffne die Tuer” computed by on the left side FFT where frequency is along the x-axis and amplitude of the frequency information of the signal along the y-axis. Second, we see in the same Figure the spectral envelope of the “Oeffne die Tuer”. The spectral envelope shows the signal amplitudes versus the frequency in the plot. On the right side we see the FFT spectrum of the envelope computed by the Hilbert transform. Now it is shown how to create the processing loop to perform envelope detection on the input signal. This loop uses the system objects that have been instantiated.

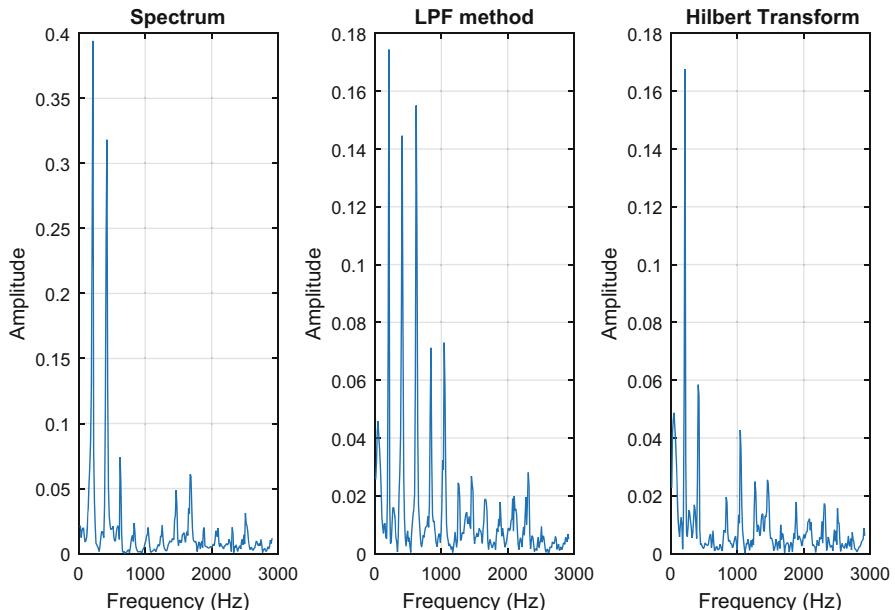
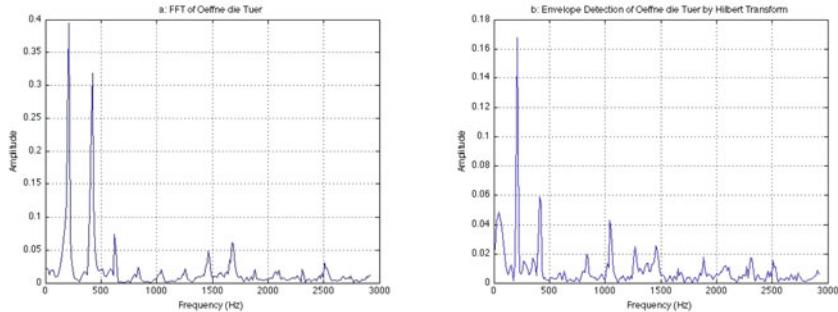


Fig. 10.3 Envelop detection by LPF method and Hilbert transform



(a) FFT of the frame of the utterance "Oeffne die Tuer" (b) The Hilbert Transform as applied to the signal depicted in (a)

Fig. 10.4 The (a) original frequency representation and its (b) Hilbert Transform representation

```

sige = abs(complex(l, step(hhilbert, sig)) + step(hdelay, sig));
sigenv = step(hlowpass, downsample(sige, DownsampleFactor));
one should plot the signals and envelopes step(hts, sig, sigenv).

```

10.3.1 Silence Threshold Selection

In many situations one does not want silence intervals, i.e. periods where no signal samples occur. These intervals are also characterized by thresholds and should sometimes be removed. Often, one removes such silence intervals immediately when they have been detected. However, as mentioned, one should select thresholds with care. This is described in Part III. For describing silence intervals one uses thresholds that have to be determined. One commonly takes for the threshold one fourth of a median of the silence envelope. There is again no precise reason for doing so. It is based on experimentation. Amplitudes below or above the threshold (depending on the use) are detected. After removing redundant elements one obtains a simpler signal. It still contains very many elements and it is still not possible to apply combinatorial computations. For a further reduction, an extraction of features is needed. In Fig. 10.5 we see the comparison of the results of the removal.

10.3.2 Pre-emphasis

Pre-emphasizing is one of the very first steps of signal processing. It is done by applying a filter. Equation 10.5 is seen as pre-emphasizing the signal and the filter in Eq. 10.5 is known as pre-emphasis filter. The constant a_{em} is discussed further below.

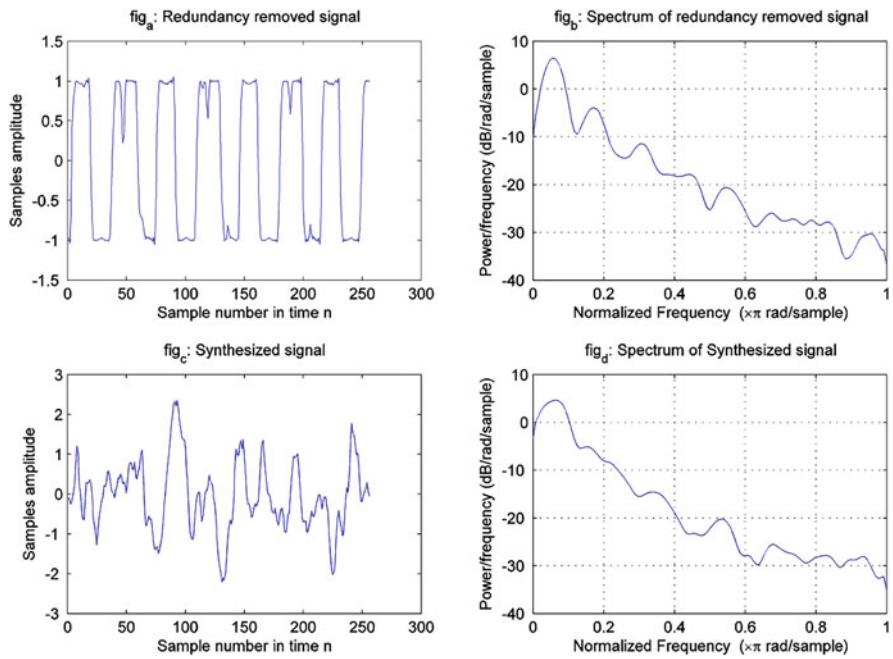


Fig. 10.5 Example of redundancy removal

$$s[n] = s'[n] - a_{em}s'[n - 1] \quad (10.5)$$

A common way of seeing the purpose of a pre-emphasis filter is to emphasize the frequency component by considering both the low and high frequency components of the signal. Equation 10.5 can be rewritten in Eq. 10.6 in the z-domain by replacing s by S :

$$s'[n - 1] \Rightarrow S'(z)z^{-1}$$

which gives:

$$S(z) = S'(z)(1 - a_{em}z^{-1}) = S'(z)H_{em}(z) \quad (10.6)$$

Therefore one gets:

$$H_{em}(z) = 1 - a_{em}z^{-1} \quad (10.7)$$

In this equation, if $a_{em} < 0$ the pre-filter acts as a low-pass filter, and if $a_{em} > 0$, the filter is a high-pass filter. This is illustrated in Fig. 10.6.

The transfer function of the pre-emphasize filter shown in Eq. 10.6 depicts just a high pass filter. The determination of the coefficient a_{em} is not based on some theory

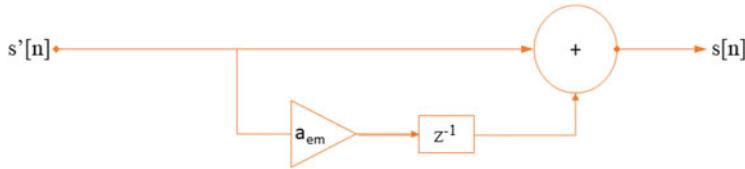


Fig. 10.6 Block diagram of a pre-emphasis Filter with a_{em} factor

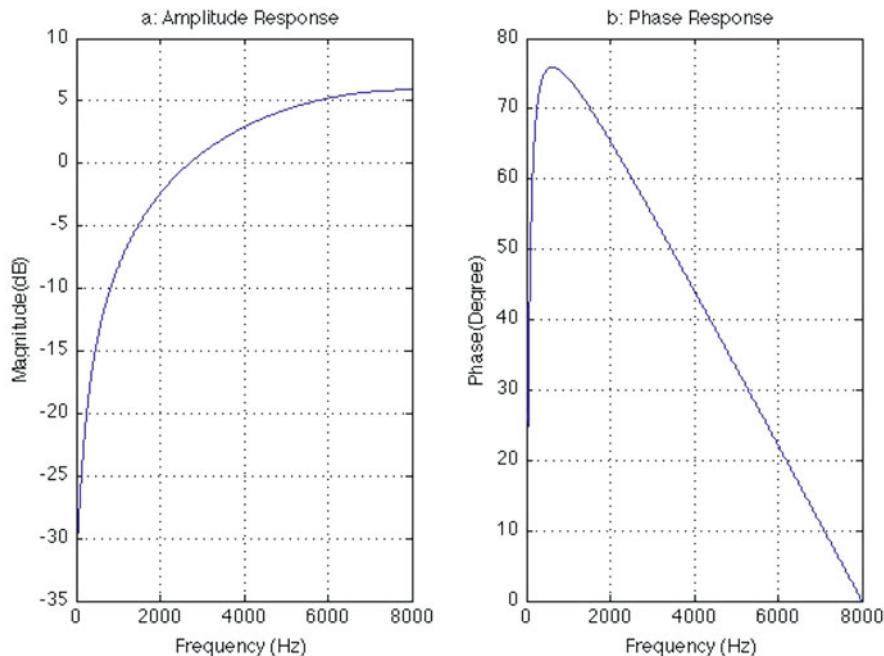


Fig. 10.7 Pre-emphasis filter characteristics given by amplitude and phase response

but again on an empirical adjustment. Hence the value is again not precisely defined and depends on the application. The frequency response of this filter increases slowly from low to high. Therefore it sets up a balance between the high and low pass frequencies. The parameter a_{em} controls the slope of the curve. Therefore, this pre-filter may be called a pre-emphasis filter. In Fig. 10.7 one sees the amplitude and the phase response of the pre-filter for $a_{em} = 0 : 97$. This response also shows that it is a high pass filter.

Next we consider the Dual-tone multi-frequency (DTMF).

Example Improving DTMF decoding with raw detector audio as the source. We consider the task to improve the DTMF decoding reliability of a device that is using a detector under the assumption that it is not emphasized. The frequencies for DTMF digits range from ~ 700 Hz (actually 697Hz) to ~ 1600 Hz, so one starts to roll off the audio at or above the highest of them, at the 1600 Hz point.

$F_c = 1600$ (frequency) presumed 10K input impedance)

$$C = 1,000,000 / (2 * \pi * 10,000 * 1600) = 0.01\mu F$$

This RC filter would deliver nearly 6 dB reduction at 3 KHz. Alternatively, choosing a slightly lower crossover frequency could be used to achieve more noise reduction at 3 KHz.

Method

In order to avoid too much “twist” in decoding, which can also prove to be a problem, we choose a crossover of 700Hz. This accommodates our desire to reduce the influence of noise, and take care of pre-emphasized user DTMF. Since most communications equipment pre-emphasizes DTMF, this makes the most sense. This option would best be used over the above, although both are included to suit differing situations. It all depends on your decoder’s tolerances, but why have “twist” if it isn’t necessary?

$F_c = 700$ (frequency)

$R = 10,000$ (10K working input to process X)

$$C = 1,000,000 / (2 * \pi * 10,000 * 700) = 0.022\mu F$$

10.4 Several Processes

Very many general signals occur when several processes are under consideration. Usually they have certain relations with each other. Each process has certain information for the receiver. The receiver has to bring them together. Because this information is not a-priori obvious the receiver has to interpret it. One can learn from one process what to do and from another process what to avoid.

When several processes occur this interpretation will be manifold what creates several problems. The information in the different processes is not always pointing into the same direction and the user has to make a choice. The processes may support each other or they may be in conflict.

From cognitive science one can take over the concept of analogy. This takes place between solutions of two problems. It is not formally defined but rather more intuitive. A consequence is that usually one says in the retrospect that two situations are analogous. We assume that the out and incoming events are partially ordered. One can interpret this ordering as a temporal one and hence we can speak about “sooner or later”. This does not mean that the ordering is linear because some events may happen at the same time.

In many situations not only one process is active. When several processes occur the following questions arise:

1. Do those processes have the same source and interpretation?
2. Are these processes related to each other and in which way?
3. In which way can they support each other?
4. As it possible that conflicts arise?
5. How can one compare different processes?

The last question is the most general one and we will pay most attention to it. The other questions are related to it. Comparisons are of particular interest if the processes are in competition with each other. This is of interest if “the best” does not exist; this situation happens quite often in medicine. This kind of comparison is closely related to usefulness and the possible consequences of the processes.

Due to the different origins of the processes one has to compare different models. Then different processes of each model have to be compared too. There are several reasons for this. For instance, when one has to determine the possible dangers of a change with respect to the health as the result of the processes. Another reason is if several processes are in competition with each other in order to find out which is the best with respect to a certain purpose. In any case, the syntactic nature of the processes is of little interest. What matters is their interpretation. Several processes together with their interpretations have been collected and are used. This is an area for case-based reasoning where comparison is central. Comparing can mean different things. For completeness one has to say in which respect the comparison should be. This depends on the aspect on which the comparison takes place. One has to say: Compare with respect to what? Then a typical question is: Which processes indicate a close relation to a very dangerous one? There may be different kinds of danger! This is a central topic of similarity cultivated in case-based reasoning. The comparison is not only with respect to equality. It gives rather degrees expressing how close two objects are. Two different signals can be very close to each other. Such comparisons require one or more parameters. For determining them one needs a computational tool that ultimately determines what more or less how close they are. One calls these tools similarity measures. Each measure has a certain focus that says to which respect one wants to compare the objects or processes under consideration. This again depends on the user. We introduce the topic in some generality.

In addition to the measures thresholds can be useful. Then one can apply the rough set method for being on the safe side and requiring additional information. This method is introduced in Chap. 14 for making decisions in the presence of uncertainty. The processes have an underlying graph structure showing the relations between the signals but the annotations are usually very complex. A general process model describes a set of processes; it allows defining specific processes. In a deterministic process the individual events are governed by (logical) rules. Otherwise one hopes for the best of having useful probabilities. This again is to a certain respect that is subjective and one needs a concept describing this. For this purpose the concept of similarity is discussed in some detail.

10.4.1 *Similarity*

Similarity is a term in everyday language. It is only a singular term and has no general plural. It has no precise definition and it is used with multiple meanings. The comparison based on similarity is not a yes-or-no question saying what is similar and

what not, but rather a similarity degree along various aspects. They are relational ones or in degrees making precise what “more or less similar” means.

In order to compare two signals or two processes from a set X one can use description of similarity as a relation or as a function. An equivalent method is to use distance measures. We first will describe and model similarities. The model can be based on relations and on functions.

There are three kinds of relational models:

1. A binary similarity predicate:

$SIM(x, y) \Leftrightarrow$ “ x and y are similar”

This is a general binary predicate.

2. A binary dissimilarity predicate:

$DISSIM(x, y) \Leftrightarrow$ “ x and y are dissimilar”

Again, a binary predicate.

3. Similarity as a partial order relation:

$R(x, y, z) \Leftrightarrow$ “ x is at least as similar to y as x is to z ”

Also, a predicate.

All these relations are understood in classical logic, i.e. they are either true or false. However, they can be weakly formulated in terms of other relations or even functions. It is more or less clear how to introduce degrees as we do it in the sequel. Now we come to the aspect of similarity as a non-fixed property. First we start with preference relations.

Definition A binary relation \geq on a set X is a preference relation if it is reflexive and transitive.

If $x \geq y$ one says in this context that x is preferred over y . An example when one buys something is “cheap” is preferred over “expensive”. This is the idea of the relation $R(x, y, z)$. In principle, as said, similarity will be interpreted as usefulness. It is clear that this depends on the situation. Now suppose y is what one ideally wants.

Definition A similarity relation is $SIMy(x, z) \Leftrightarrow$ “ x is at least as similar to y as z is”.

A derived preference relation is: Then we would prefer x rather than z . This relation is provided by the user and is not universal. It also gives a precise meaning to the concept of similarity. Two objects are similar if one cannot distinguish them for decisions. This is further extended to the use of functions. First we introduce distance functions and compare them to similarity functions.

Definition

- Distance function are the form $d(x, y) : \mathbf{X} \times \mathbf{Y} \rightarrow [0, 1]$.
- Similarity measures are of the form $sim(x, y) : \mathbf{X} \times \mathbf{Y} \rightarrow [0, 1]$.

Distance functions tell us how far are the objects from each other, while similarity measures say how close they are. Instead of the unit interval as the range one can also take an extension, for instance the set of all non-negative real numbers. Intuitively the functions are two ways that are essentially equivalent. The meaning of the

function sim is the one from the derived preference relations. The addition coming from the functions is that one can now express degrees. There are different ways to compute the functions from each other. The computations observe the relations but not all other properties. In addition, distance functions and similarity measures are more or less obvious from the application view but they have differences in their computational difficulty.

There are very, very many of such functions in the same way as there are many preference relations. This reflects the intuitive feeling that similarity is not unique. As examples one can pick a range of them where we discuss bio-medical applications in Chap. 24 in Part III of the book. Such a function assigns values to elements where lower values may be preferred. We make the basic idea precise for the best choice in the following concept of nearest neighbors.

Definition Suppose we have a set U , a subset X of U , a similarity measure sim and an element x of U . A nearest neighbor of u to x in X is defined by $N(x, X) \leftrightarrow \forall y \in X : sim(x, u) \geq sim(x, y)$.

Observe that there may be more than one nearest neighbor. This depends on the similarity measure sim . In a very extreme situation every object may be a nearest neighbor of every other object. A requirement is that different formulations of the same similarity concept in a situation should preserve the nearest neighbor relation.

10.4.2 The Local-Global Principle

The local-global principle can be defined for arbitrary objects; it is known from object oriented programming. We have:

- Each Object is constructed from components A_i by construction process $\mathbf{C}(A_i | i \in I) = A$.
- Here \mathbf{C} is the construction operator. Such construction processes apply to all considered types of description even when uncertainty is involved.
- The occurring functions can be fuzzy membership functions.
- In this context the objects A_i are local objects and A is a global object. Hence local and global are relative notions that depend on the situation.

The goal is to say something about a global object on the basis of local objects. The local-global principle says that other definitions should go from local objects to global one.

Questions arising are:

- What can one say about the construction operator \mathbf{C} ?
- Are there normal forms for \mathbf{C} ?
- What are the consequences of properties of \mathbf{C} ?

These questions cannot be answered uniformly. The answers describe to some degree the nature of the situations. Now we use the principle for similarity measures.

There are *local measures* sim_i on the domains of attributes A_i and there is some amalgamation function F such that for $a, b \in U$, (U being the universe under consideration), $a = (a_i | i \in I)$, $b = (b_i | i \in I)$ one has $sim(a, b) = F(sim_i(a_i, b_i) | i \in I)$.

The measure sim is called the global measure. The function F is difficult to establish because it incorporates the interplay between the different components. The local measures sim_i compare values of individual attributes and sim compares the objects from a global point of view. For signal processes the local view is on the signals and the global view is on the whole processes. Processes can be again combined to give other processes.

10.4.2.1 Similarity and Utility

The inequality $sim(x, u) \geq sim(x, y)$ is a preference relation between u and y . It is obvious how to express this in terms of distances. The intention is now to select an element from X that is the closest to x ; “closer is preferred”. That means the nearest neighbor is the one we want most. However, there may be more than one nearest neighbor, depending on the similarity. One usually prefers the object that is most similar to the “best” object. This is standard for processing objects. This shows that similarity expresses utility. For signal processes this is important for instance in the medical application that we will consider in the next part. This also shows the contrary hypothesis: One should avoid the least similar choice. Then one could use a distance measure rather than similarities.

10.4.2.2 How to Define a Similarity Measure

The similarity measure is a central point in similarity reasoning and often it is difficult to define it appropriately. This means mainly that it corresponds to the intention which is the utility. Persons have a better feeling to what is useful than to what is similar. Special forms of difficulties arise for stochastic processes, in particular if the probabilities are unknown. There are quite many methods for defining a similarity measure. Here we suggest a learning method. It is in principle not very complex. The purpose is to simplify the purpose and not to admit big errors. We will outline this next. This goes step-wise. The start is to get a primitive and not very accurate measure. The following tasks for the first primitive measure have to be done:

1. Define the language of your topic with basic attributes.
2. Define a local-global principle.
3. Select a set of examples.
4. Select attributes that are not quite important and give them the importance $\omega_i = 0$
5. Select pairs of examples that are not similar and give them a very low similarity sim_i .

6. Select attributes where you think they are important for the dissimilarity in this language and give them an importance between 0.5 and 0.9.
7. For the other attributes give them a measure sim_i where you think it is adequate.
8. The obtained values for the attributes and importance define a similarity measure sim between x and y :

$$sim(x_i, y_i) = \sum_i^n \omega_i sim(x_i - y_i) \quad (10.8)$$

In this way one obtains a similarity measure sim which is very simple and not very correct but the errors are still small. After that an iteration takes place in the form of learning:

1. For the initial measure the primitive one obtained above is taken. For a question q the measure arranges the solution of the similarity question in a sequence $c_1 \geq \dots \geq c_n$.
2. Now the correct answer from an expert or the user might be different. For example one might get $c_5 \geq c_1 \dots \geq c_3$. This correct answer is now used for improvement. The similarity measure is changed such that this sequence is obtained.
3. This step is iterated. There are two ways of representing probabilities and both give rise to a learning goal. The first one is relational: The relational is $P(A) > P(B)$. The second one is functional: The function is $P(A)$. To learn a relational version was just discussed. We also said how to obtain a similarity measure from the relations. The same method works here.

DTW We start with the description and computation of similarities between linearly ordered processes. Often the ordering comes from time and hence one calls them time series. In order to compare two time series one mostly uses some distance measure. A popular one is DTW (Dynamic Time Warping). DTW(x, y) denotes the similarity between the processes x and y . As just mentioned, one compares usefulness. In the bio-medical applications one can express this for instance as “more or less healthy”. The special aspect of DTW is that it is close to the development of utility. DTW is expressed in terms of distances. One starts with a distance measure between the elements of the two processes. Often this is the Euclidean distance.

The idea of DTW is that DTW(i, j) is the minimal cumulative distance when mapping the first i values of the first time series to the first j values of the second time series. Because one uses tests for comparison we use this kind of terminology. The DTW measures the distance between two tests. Mostly one is a query test process and the other one is a reference process among the ones to which we want to find the best match. This is done for optimization problems. The similarity (or equivalent distance) measurement allows comparing two processes. One wants to find out which test vector is closest to the reference vector. We consider a reference

vector and test vectors coming from a training set. This similarity for processes is now measured in two inductive steps.

Definition Local similarity measure on the individual signals: Here the similarity of each signal of the reference and a test vector is investigated. The local distance is measured using the Euclidean distance presented in the Eq. 10.9, where i, j in x_i and y_j denote individual signal events in the reference and test vector. This formulation is shown in equation below.

$$d(x, y) = \sum_{i=1}^k (x_i - y_i)^2 \quad (10.9)$$

A Global similarity measure D on the whole process - Here the similarity of the signals between the reference and the test process are investigated. Each point in the matrices is marked as a node.

$$D(i; j) = d(i; j) + \min[D(i - 1; j - 1); D(i - 1, j); D(i, j - 1)] \quad (10.10)$$

There the global DTW path is $D(i, j)$. This is computed up to node (i, j) where the local distance at (i, j) is given by $d(i, j)$. The entry at coordinates (i, j) in the DTW frame is the value of the minimum cost mapping through a cost matrix which is a weighted Euclidean distance matrix. The distance for aligning empty sequences is initialized as $DTW(0, 0) = 0$, and the distance for fully aligning the two processes is computed at $(i - 1, j - 1)$. DTW computes the similarity measure by the sum of Euclidean distances at each alignment point, plus the cost of transitions between such pairs.

In the Fig. 10.8 the distance is measured horizontally, vertically, and diagonally. The horizontal, vertical and diagonal distances are denoted by $(i - 1, j)$, $(i, j - 1)$, and $(i - 1, j - 1)$. Thus, in the DTW transitions, the three options to the next mapping are: (1) move to the next element in the first time series only, (2) move to the next element of the second time series only, or (3) move to the next element in both time series. The shortest cost path from $(1, 1)$ to (M, N) is usually nearly diagonal. The cost matrix is multiplied with some weight factor which is 1 in this example for a diagonal step. This also assists to prevent any skipping or the default shortest minimum cost path.

To achieve insight into the process recognition performance of DTW, the recognition results have to be analyzed thoroughly. As examples one checks the reflexivity and symmetrical property in the distance measurement technique. One may complete this test by measuring DTW distance using the same speech features. The zero results in this test fits a correct DTW technique when using Euclidean distance measurement in the speech sequences here.

The statistical DTW distribution score can use several probability distributions such as gamma, exponential, log normal, normal, or Weibull distribution. The DTW distance scores in the listed words are varying. The probability distribution

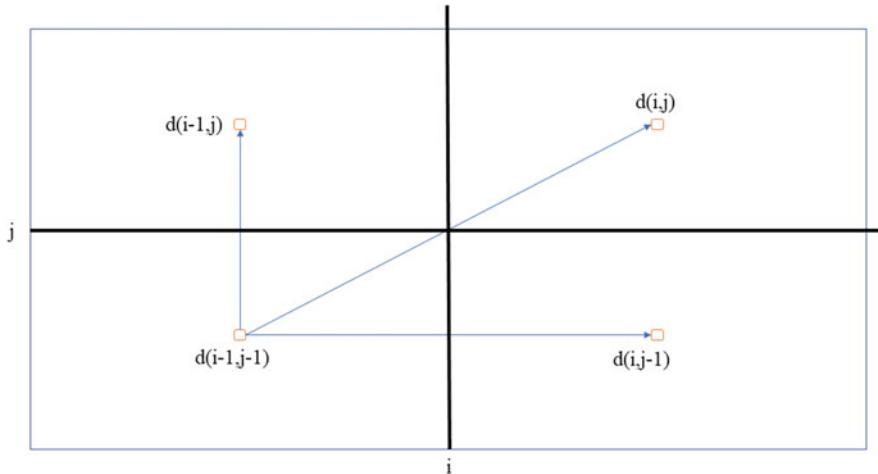


Fig. 10.8 DTW frame depicting allowable paths

determines the best fitting distribution and the estimation of the parameters for that distribution. A probability distribution is characterized by location and scale parameters; these parameters are used in modeling applications. When considering the shape of a distribution of scores, it is useful to measure the skewness and kurtosis. Kurtosis is any measure of the “peakedness” of the probability distribution of a real-valued random variable. The term skewed is used to refer to something out of line. A distribution with an asymmetric tail extending out to the right is referred to as positively skewed.

One way to analyze the DTW result is by calculating the probability distributions using the mean of the DTW distance scores of different tests, minimal DTW distance score of these tests, and the variation of the distance by computing the range (difference between maximum and minimum value of DTW distance score) of the DTW distance score at each testing.

As a measure of distortion between the source sequence x and the template we can use the distance $d_M(x, y) = (x - y)^T A(x - y)$. It can be thought of as a weighted version of the Euclidean distance. If we consider x to be a multivariate Gaussian with mean y and covariance matrix A^{-1} , then we can relate this distance to the probability of x :

$$p(x) = \frac{1}{|2\pi A|^{\frac{1}{2}} e^{\frac{1}{2}(x-y)^T A(x-y)}}$$

$$D_M(x, y) = -\log(p(x)) + c \quad (10.11)$$

$$A = [a_{ij}] \quad W = [\omega_{ij}]$$

$$a_{ij} = \frac{e^{-\omega_{ij}}}{\sum_{j'} e^{-\omega_{ij'}}}, \forall i \text{ and } \sum_j a_{ij} = 1$$

$$\omega_{ij} = \infty \Leftrightarrow a_{ij} = 0$$

The scale parameter is the measure of the variation of the samples from location parameters. The effect of a scale parameter greater than one is to stretch the probability distribution function (PDF), and if the effect of a scale parameter less than one is to compress the PDF. The compressing approaches a spike as the scale parameter goes to zero.

10.4.3 HSM Similarities

The HSM similarities are in the first step not obtained by learning methods so far. Given a reference process x and a test process y the similarity between the two is obtained by the probability:

$$sim_{HSM}(x, y) = Prob(x, y)$$

This assumes that the probabilities are known. If not, there is still a learning goal, namely to learn these probabilities. This was discussed above. In general the methods of learning stochastic processes apply here to HSM also.

A more refined measure is DTW introduced above. Now we show how DTW differs from HSM. In both, HSM and DTW, one tries to find the most suitable word for a test word A_v in the given vocabulary. However, they measure this in different ways. DTW take the smallest distance where HSM takes the highest probability. The difference between DTW and HSM is formally expressed in Eq. 10.8 and in Eq. 10.11. In Eq. 10.8 and in Eq. 10.11 the signal is denoted by W . Now we show how to apply DTW to the recognition problem in two versions.

$$W^* = \operatorname{argmin}_w \operatorname{distance}(A_v, w) \quad (10.12)$$

$$W^* = \operatorname{argmax}_w \operatorname{Prob}(w, A_v) \quad (10.13)$$

For both choices represented with Eqs. 10.12 and 10.13, one needs an optimization for the nearest neighbor search, that is clarified next.

10.5 Conflict and Support

Conflicts and supports have three meanings.

- The first one assumes that there is an ordering on the signals. It specifies whether the ordering coming from one process points in the same or in the opposite direction as compared to the one from the other process.
- The second is that the information of one process is the same or conflicting with than the one of the other process.
- The third meaning is an extended semantic when one looks at the resulting actions. These may support each other or may be in conflict, too.

Such relations among stochastic processes are mostly considered when human decisions are involved. This is not the only topic here. We accept the processes as information and not as demands. A demand can only be the human consequence of a process.

For signal processes one has to focus on the detection of conflicts and cooperation at the network level. The conflict with processes indicates the different properties they have. This can take place when a reward of interest or penalty for processes derived from the basic underlying process is considered. Typical examples are state-dependent impulse reward measures.

The support of processes means that the properties of processes make a recommendation in the same direction. In general, both, conflict and support are handled on the level of symbolic properties of signal processes and not for the signals themselves. That means one has first to interpret the signal processes. An exception is if one considers terms referring to the signal-noise relation as the signal to noise relation. For this one takes the quality of noise removal into consideration. That is part of the quality of process generation and can be relevant.

10.6 Examples and Applications

Examples and more elaborate applications are provided in Part III. That means for comparisons of processes not only the contained information but the processes themselves are compared. Above we discussed similarity measures on processes. There are very many quite different applications, for instance in economics, medicine or environment. In some sense they follow all the same principles. The differences are not only in the purpose but more relevant in the kind of stochastic processes. They contain a set of different observed time sequences to which one wants a comparison in order to see what might be happening in the future. Often, there it is necessary to compare such sequences and processes. This is because one is interested in terms of developments for which the utility is usually defined. There one mainly uses DTW similarities for the measures. A very general application that occurs in many situations is prediction. For other applications as for instance in

weather forecast detecting the message of a time sequence is relevant. There are different reasons for this. One is that the message is not of interest for the present purpose but for something in the future. Another one is that it is relevant but one does not want that it becomes public. These are of a general character and will be discussed further in Part III.

10.7 Predictions

A frequent general application is prediction that is a part of many applications. For processes one collects signals from the past and wants to know what comes next. This may also concern a sub process. One raises the questions:

- Is the prediction precise? In which way?
- Does it indicate an improvement or a drawback?
- Has one to do some action right now or later as a response?

This refers to the meaning and its exactness of the measurements and their results. The last question expects an answer and possibly an action. One has a lot of inputs that interrelate with each other. This creates a complex task. Some applications of predictions come from examples. They are from many domains. Frequent examples are from economics as well as bio-medical areas and have many uses in business and medicine itself. They are all based on certain signals and signal processes.

We make at this place some short remarks about predictions only. In Part III such examples are discussed in more detail. A major application there is speech. The origin is a speaker or an organism and the receiver wants to understand it. The receiver can be a human or a machine.

A basic question is what the meaning of understanding is. This is different for humans and machines although it has some commonalities. In general one expects from understanding the right response. For humans this response can be a right answer or an intended action. We neglect the term “deep understanding” that is sometimes connected with understanding from humans. For machines it is just the action (that could also be an artificially created speech). In medical applications the origin of the signals is a living body and a medical agent wants to know what comes next for adjusting the therapy. Also the seismic topic is discussed below in Part III where the origin is the earth. As an example one might ask: Do we expect an earthquake? All these applications demand a special treatment for predictions in the involved stochastic processes.

10.8 Background Information

General

An early influential book on the early main topics of this chapter is “Stochastic processes” by J.L. Doob from 1953, (Doob 1953). One approach to stochastic processes treats them as functions of one or several deterministic arguments (inputs, in most cases regarded as in time) whose values (outputs) are random variables: non-deterministic (single) quantities which have certain probability distributions. The book Vaseghi, S.V. (2007) contains many aspects of stochastic processes. In addition it presents a comprehensive discussion of multimedia that will be presented in Part III.

History

The Markov process was named after Andrey Markov; in the literature it is often identified with Markov chains where the signals are linearly ordered. The classical theory of Markov chains studied fixed chains, and the goal was to estimate the rate of convergence to stationarity of the distribution at time t as $t \rightarrow \infty$. In the past two decades, as interest in chains with large state spaces has increased, a different asymptotic analysis has emerged. The Markov chain state space does not have an established definition. The term may refer to a very general process (Fink 2003). Familiar examples of processes modeled as stochastic time series include signals such as speech, audio and video, medical data such as a patient’s ECG, EEG, blood pressure or temperature and in addition seismic processes. One finds such applications in Part III.

In the theory of stochastic processes, the Karhunen-Loeve (Loéve 1978) description gives a representation of a stochastic process as an infinite linear combination of orthogonal functions, analogous to a Fourier series representation of a function on a bounded interval. A contrast is to a Fourier series. There the coefficients are real numbers and the expansion basis consists of sinusoidal functions (that is, sine and cosine functions). The coefficients in the Karhunen-Loeve theorem are random variables and the expansion basis depends on the process. The importance of the Karhunen-Loeve theorem is that it yields the best such basis in the sense that it minimizes the total mean squared error.

The Hilbert transform (named after the mathematician David Hilbert) is a linear operator which takes a function f and produces another function F with the same domain. It is a basic tool in Fourier analysis and signal processing. It is used to derive the analytic representation of a signal. The goal is to find the possibly smallest area which includes the graph of the given function. For the implementation and other details about the Hilbert transform we recommend Losada (2008). An alternative to the discussed redundancy removal can be found in Allen and Mills (2004).

The early references of DTW are in the speech recognition literature that we discuss below; see for instance Myers and Rabiner (1981). An improved version is FastDTW, Salvador and Chan (2004). An implementation in Java is available, see <https://cs.fit.edu/~pkc/papers/tdm04.pdf> Markov processes (also called Markov models or Markov chains) are named after the mathematician Andrey Markov. For

more information see Kaelbling et al. (1998) and Ibe (2008). Medical applications are in Sonnenberg and Beck (1993). Hidden Markov Models have first been chosen to model Markov chains with unknown probabilities. We used an extended version to cover all stochastic processes, not just Markov models. The inner states of the system are unknown but one sees for each state certain outputs, called emissions. From these one can generate some statements about the hidden probabilities. In order to make a step in this direction again learning is applied (Lanchantin and Pieczynski 2005). The mathematics behind the learning for HSM was developed by L. E. Baum and co-workers. In (Baum and Petrie 1966), one finds much information about the involved mathematics.

Suggested

Applications of HSM to pattern recognition are in Satish and Gururaj (2003) and Myers and Rabiner (1981). More about pre-emphasis is in Chen et al. (2009) and Rabiner and Juang (1993). Case based reasoning is some time on the market and has success in the market. A central point is the concept of similarity measure. It started with making use of experiences. Two experiences are relation if they are close to the similarity measure. The reference Richter-Weber (Richter and Weber 2013), contains a comprehensive study that contains a discussion of similarity, including many examples and applications. It includes both, theoretical and practical views. It categorizes applications and gives hints to choose useful similarity measures. An algebraic approach to conflict and support is given in Brinksma et al. (1995). For quality aspects of methods there are many occasions in this book. These aspects are part of the intended utility.

10.9 Exercises

Exercise 1 Suppose there are binary signals with only the digits 0 and 1. It is supposed to transmit one of these digits through several stages. At every stage there is a probability p that the digit that enters this stage will be changed when it leaves and a probability $q = 1 - p$ that it won't. Form a Markov chain to represent the process of transmission by taking as states the digits 0 and 1. What is the matrix of transition probabilities?

Exercise 2 An absorbing state is a state that, once entered, cannot be left. An absorbing Markov chain is a one in which every state can reach an absorbing state. Consider a Markov chain with two states 0, 1. Suppose there are two probabilities $p = 1$, and $q = 0$. For which values of p and q do we obtain an absorbing Markov chain?

Exercise 3 Consider the process of repeatedly flipping a fair coin until the sequence (heads, tails, heads) appears. Model this process by using an absorbing Markov chain and find the transition matrix?

Exercise 4 Suppose three time series are given:

1. $x = (3,4,5,3,4,4,6,4)$
2. $y = (4,4,4,5,5,5,5,2)$
3. $z = (2,3,5,3,2,2,3,2)$

What is the closest series to x with respect to DTW, y or z ? Prove your choice?

Exercise 5 Give an example of a stationary process which is not ergodic. A process is defined to be ergodic if it is not changing its statistics, such as mean, standard deviation, skewness, kurtosis, etc?

Exercise 6 Let X be a binary symmetric first-order Markov process as in the previous question, with a transition probability α , $0 < \alpha < 1/2$. Find an output Z of a memory-less binary symmetric channel.

References

- [Allen2004] Allen R. L., Mills D., Signal Analysis: Time, Frequency, Scale and Structure. Wiley-IEEE press, NJ, 2004.
- [Sonnenberg1993] Sonnenberg F.A., Beck J. R., Markov models in medical decision making: a practical guide. Med Decis Making. National Library of Medicine, NIH, USA, 1993.
- [Satish2003] Satish, L., Gururaj BI, Use of hidden Markov models for partial discharge pattern classification. IEEE Transactions on Dielectrics and Electrical Insulation, 2003.
- [Rabiner1993] Rabiner, L.R., Juang, B.-H., Fundamentals of Speech Recognition. Prentice Hall, New Jersey, USA, 1993.
- [Baum1966] Baum, L. Petrie, E., Statistical Inference for Probabilistic Functions of Finite State Markov Chains. The Annals of Mathematical Statistics 37 (6): 1554–1563, 1966.
- [Doob1953] Doob, J. L. Stochastic Processes. Wiley, 1953
- [Fink2003] Fink, G. A., Mustererkennung mit Markov-Modellen: Theorie, Praxis, Anwendungsbiete. Teubner, 2003.
- [Kaelbling1998] Kaelbling, L.P., Littman, M.L., Cassandra A.R., Planning and acting in partially observable domains. Artificial Intelligence (Elsevier) 101. 13(4):322–38, 1998
- [Lanchantin2005] Lanchantin, P., Pieczynski, W., Unsupervised restoration of hidden non stationary Markov chain using evidential priors, IEEE Trans. on Signal Processing, Vol. 53, No. 8, 2005.
- [Richter2013] Richter, M. M., Weber, R. O., Case Based Reasoning. A Text Book. Springer Verlag, 2013.
- [Loeve1978] Loéve, M., Probability theory. Vol. II, 4th ed. Graduate Texts in Mathematics 46. Springer-Verlag, 1978.
- [Myers1981] Myers, M. S. and L. R. Rabiner, A comparative study of several dynamic time-warping algorithms for connected word recognition. The Bell System Technical Journal, 60(7):1389–1409, September, 1981.
- [Ibe2008] Ibe, O.C., Markov Processes for Stochastic Modeling. Elsevier, 2008.
- [Brinksma1995] Brinksma, E., Katoen, J.-P., Langerak, R., Latella, D. (1995): A Stochastic Causality-Based Process Algebra. The Computer Journal 38 (7): 552–565, 1995.
- [Hasegawa2005] Hasegawa-Johnson, M. Lecture 3: Acoustic Features. <http://www.ifp.uiuc.edu/>, June, 2005.

- [Salvador2004]** Salvador, S. and Chan, P., FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. KDD Workshop on Mining Temporal and Sequential Data. Pages :70–80, 2004.
- [Losada2008]** Losada, R.A., Digital Filters with Matlab. The MathWorks, Inc., 2008
- [Chen2009]** Chen, S.F., Picheny, M., Aramabhadran, B., Advanced speech recognition. IBM T.J. Watson Research Center Yorktown Heights, NY, USA, 2009.
- [Vaseghi2007]** Vaseghi, S.V. (2007), Advanced Digital Signal Processing and Noise Reduction, John Wiley and Sons, Ltd, 2007

Chapter 11

Feature Extraction



Overview

Signal processes can be very complex and may contain a huge number of individual signals. This makes combinatorial computations for the determination of properties difficult and often impossible. The use of features is a common way to deal with this problem. Arbitrary signal features are (short) real valued vectors. They describe the processes to some extend and can be manipulated to reduce or increase computational load as needed.

If short feature vectors could be obtained without losing relevant intended information, they can significantly speed up computation with little loss. However, the remaining problem is when features may not contain the original information anymore or they only have reduced information about the processes. There one has to consider several possible features that preserve different kinds of information. Feature extraction has to take care of this.

The currently most popular feature vectors are the [Mel-Filter Cepstrum Coefficients](#) (MFCC) and the [Perceptual Linear Prediction Cepstrum Coefficients](#) (PLP). In the extraction methods MFCC and PLP the signal decomposition and spectral analysis is followed by a lapped transform based on FFT. For the problem of abrupt discontinuity the decomposition is followed by the lapped transform. In some of the applications in Part III, MFCCs are used. The details of feature extraction are discussed in Part III where we consider applications like speech processing deploying speech recognition. Here we introduce the basic idea.

A major point will be that features are not unique but their choice depends on the user. This concerns the information contained in the features: The user determines which information is of interest. Below in the applications we will make these tasks more precise by examples.

A basic expectation regarding features is their efficient computation. This is because the whole purpose of features is governed by efficiency.

11.1 Feature Extractions

The transformation of signals into feature vectors is called feature extraction. The feature extraction produces features. There exist several techniques leading to different features representing different information units. Relevant techniques have some assumptions that will be considered below. However, there is no uniform theory covering them. This has several reasons. It starts with the fact that there is no precise and uniform description of the whole intended purposes. These elements depend almost always on experience within the various domains. As a consequence the various constants occurring in the feature extraction equations are almost always numbers coming from experience. More precisely the high level purpose of the feature vectors is the following:

- Extract important information from a long sequence of signals.
- Reduce redundancy in the signal.
- Keep important information of interest.

All three demands contain non-precisely defined elements that are given from situation to situation by the user. The features are extracted from each frame of the signal. Three preliminary steps in prediction applications are as shown in Fig. 11.1. They point into the same direction as the whole feature extraction but are just first steps.

One starts with the signals and breaks them up into pieces small enough and one hopes that one can directly use them. This is mostly not the case and then the proper extraction has to take place. These steps occur in smaller segments and blocks. However, they say nothing about the specific extraction methods. There are

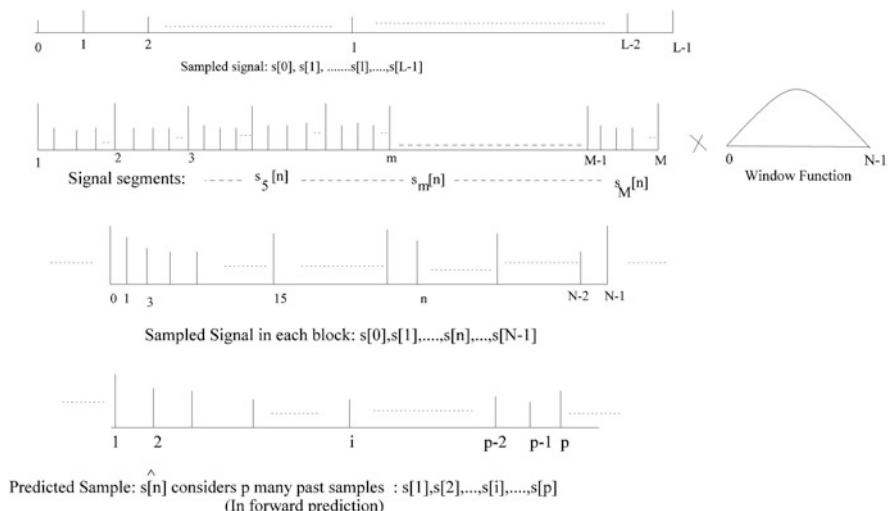


Fig. 11.1 Main processing steps before feature extraction and prediction

quite many such methods. They depend on the given environment and the purpose chosen by the user. This is discussed next.

11.2 Basic Techniques

General concepts and technologies concerning feature extraction are discussed next. This includes the structure and the content of the features.

11.2.1 *Spectral Shaping*

In spectral shaping the signal is prepared for the spectral analysis. This elaborates certain needed details. The spectral shaping has several sub-steps:

- Decompose the signal in a dyadic manner. This is standard and will not be detailed here.
- Spectral shaping
- Pre-emphasis
- Windowing the signal
- Apply a folding operator
- Unfolding

We will discuss the steps in the following, with the goal to decompose the complex signal process in such a way that certain actions can be performed. In addition one has to take care that the action can be reversed. The spectral shaping can be done in different ways that are discussed now.

11.2.1.1 Pre-emphasis

The signal is pre-emphasized by using a pre-emphasis filter. The pre-emphasized filter is divided into some segments for its detailed analysis. Segments are intervals in a process. The segmented signals are smoothed by using some window function which can be fixed or adaptive. Thus these segmentations are used in the sequel. The folding generally fixed or adapted as categorized below. The folding operator is discussed below. First we look at redundancies. In principle they are unnecessary and should be removed. An example about the signal redundancy removal is shown in Fig. 11.2. This figure shows the result of handling a signal process coming from speech signals by using redundancy methods. One realizes the data preparation and data pre-emphasize by pre-filtering shown in Fig. 11.3. This reduces the redundancy and increases the smoothness of the data-samples in two steps:

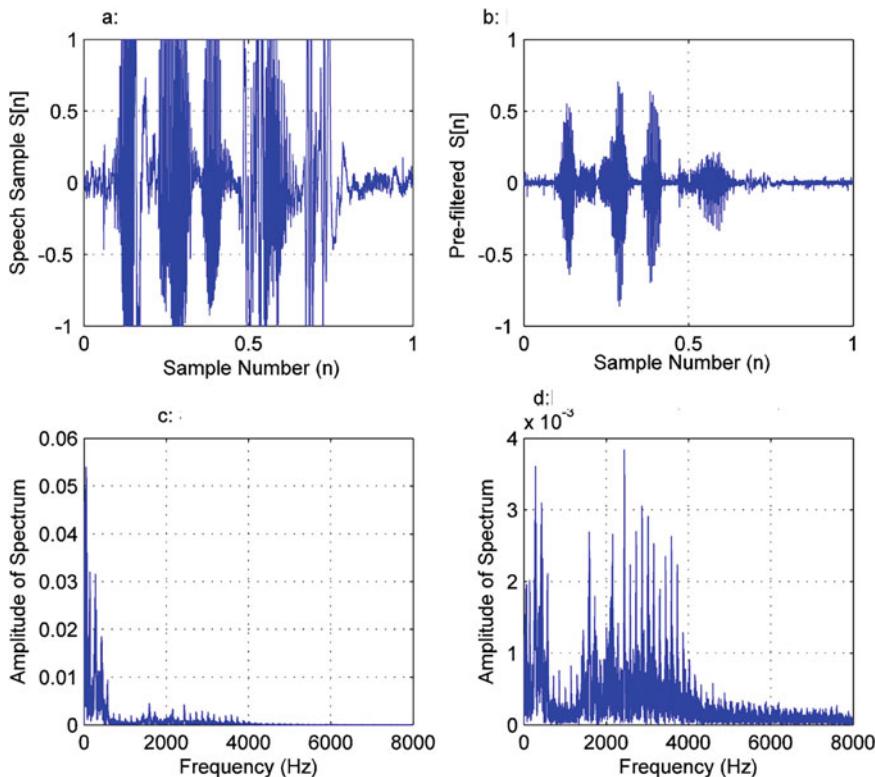


Fig. 11.2 Pre-filtering and pre-emphasized signal. (a) Redundancy removal speech. (b) Prefiltered redundancy removal speech. (c) Speech spectrum. (d) Prefiltered emphasized spectrum

- i. The reduction step by redundancy removal,
- ii. Pre-emphasizing the signal by applying pre-emphasis filter.

In the reduction step one removes silence and then one uses the pre-filtering to smooth the data. One is concerned about building an adequate model. To build such a model related to a process model system, one needs several hundreds of sample data. On the contrary, for a very general system one needs several thousands of data. A higher collection of for instance several thousands of data is not feasible. Therefore we consider approaches and we will build a special system with minimal amount of data. The special form depends on the type of application. For speech recognition it could mean that one looks at a speaker dependent function. The local cosine transform reduces the blocking effects and smoothes the signal. The principle of overlapping between the adjacent blocks is used in the next steps.

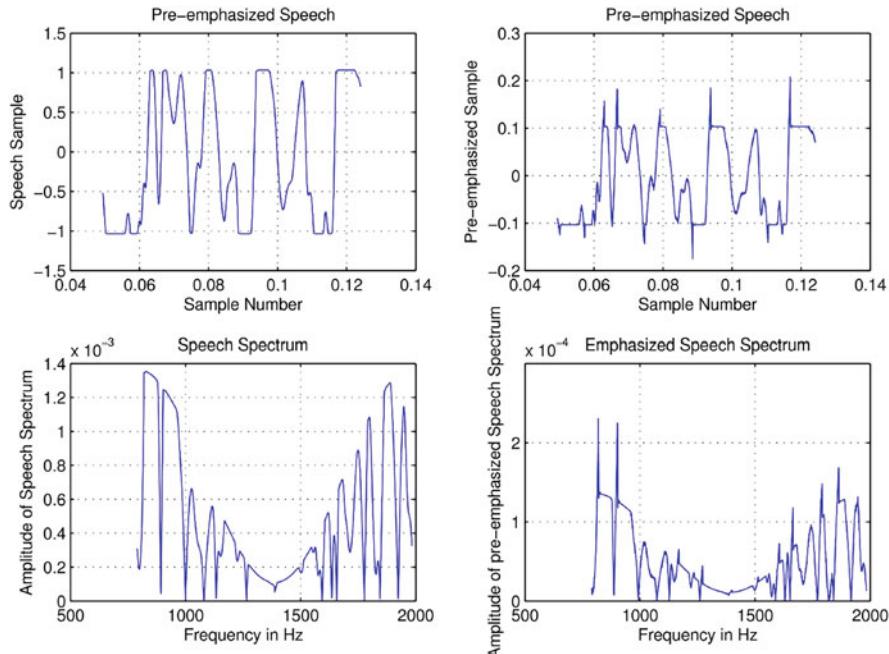


Fig. 11.3 Frame of Speech with corresponding Spectrum, and its pre-emphasized versions

11.2.1.2 Spectral Shaping by Windowing

The letters here have a slightly different meaning than in the earlier notation. The signal is framed into m blocks by applying a window function of length N in such a way that each signal of the m blocks has N many samples where $m = 1, 2, \dots, M$ and $n = 0, 1, \dots, N - 1$. This segmented signal is denoted as $s_m[n]$: this signal is a frame (e.g., collection of samples). There the windowing is applied. The formulation of result of the framed signal by the windowing is shown in Eq. 11.1 in the next section. The length of the window is equal to the signal length N . Thus the framed or the windowed signal $s_m[n]$ is the multiplication of the signal $s[n]$ by the window function $w[n]$. A shift is applied afterwards. Now:

- By multiplying the signal $s[n]$ by the window function $w[n]$ one attains the signal frames denoted as $s_m[n]$.
- Shift r is the shift in samples between adjacent segments and $r - N$. This shifting is sometimes known as overlapping. This can take place at each one half or two third of the length of the signal where the length of the signal is assumed to be the same as the length of the signal block. The shift is essential to remove the discontinuities obtained by the framing.

11.2.1.3 Windowing the Signal

When decomposing the signal into blocks in which Fourier extraction takes place then the signal is first decomposed into segments. The problem is now that there are discontinuities in the transfer at the boundaries. The result would be that the Fourier coefficients change at the boundaries and one does not have a uniform representation on the blocks. Therefore one wants to avoid this. The technical details have been introduced in Part I. Windowing is a special technique in order to avoid problems arising in simple decomposition. This is done by first introducing a window function. The signal is framed into m blocks by applying a window function of length N in such a way that each signal of the m blocks has N many samples where $m = 1, 2, \dots, M$ and $n = 0, 1, \dots, N - 1$. This segmented signal is called $s_m[n]$ in Eq. 11.1, and also is known as a **frame**.

The list of steps in feature transformation including spectral shaping and analysis is shown in Fig. 11.4. The framed signal or the windowed signal $s_m[n]$ is the multiplication of the signal $s[n]$ by the window function $w[n]$. Multiplying the signal $s[n]$ by the window function $w[n]$, we attain the signal frames. This shows how to apply DTW to the recognition problem in two versions denoted by $s_m[n]$ for $m = 1, 2, \dots, M$ and $n = 0, 1, \dots, N - 1$, where $r \leq N$ is the shift of the samples between adjacent segments. Next, we illustrate this.

$$s_m[n] = s[n + mr]w[m] \quad (11.1)$$

Later on, the $s_m[n]$ is used for the power spectrum computation (see Part III). Further we give an overview concerning the interplay between spectral shaping and spectral analysis.

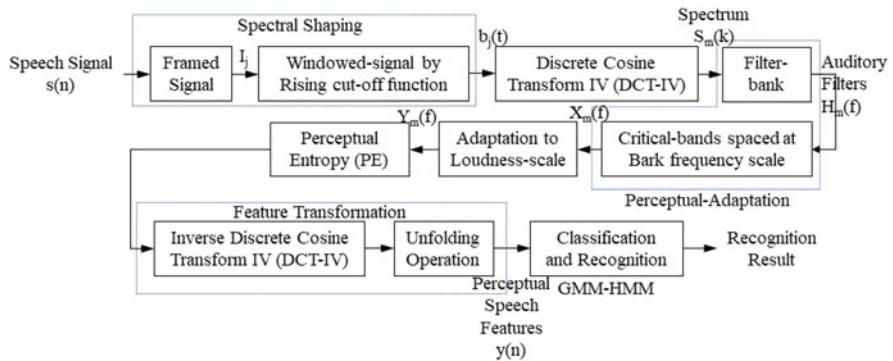


Fig. 11.4 Spectral shaping and analysis

11.2.1.4 Folding

A central operation in the procedure is now to split each interval into two smooth overlapping intervals and to construct a basis for each one. These will then be iterated and integrated to obtain a uniform basis. The smoothness is obtained by a rising cut-off local cosine trigonometric transformation. Then, this is used for the integration of the two intervals, followed by the task of unfolding. One uses smooth cut-off functions to split the signal and to fold the overlapping sections back into segments in such a way that the orthogonality is preserved. To obtain better frequency localization the signal is multiplied by a smooth window function which uses the local sine and cosine bases consisting of sine or cosine multiplied by smooth compactly supported Bell functions.

Equation 11.2 shows an inner product of the local cosine basis using the folding operator and DCT-IV. There the segment $s_j[n]$ starts with the multiplication of the window $b_j(t)$ and the signal $s[n]$. The multiplication is folded at the edges using the folding operator and then DCT IV is then applied on the folded result giving the disjoint window segments $s_j[n]$.

$$s_j[n] = \begin{cases} b_j[n]s[n] + b_j[2a_j - n]s[2s_j - n] & \text{if } a_j \leq n \leq a_j + r \\ s[n] & \text{if } a_j + r \leq n \leq a_{j+1} - r \\ b_j[n]s[n] + b_j[2a_{j+1} - n]s[2a_{j+1} - n] & \text{if } a_{j+1} - r \leq n \leq a_{j+1} \end{cases} \quad (11.2)$$

We denote this as $s_j[n] = b_j[n]s[n]$.

In Fig. 11.5 we see an example of the folding. Later on, unfolding can reconstruct back the original signal.

11.2.1.5 Feature Transformation and Unfolding

Folding is an intermediary process useful for analysis and manipulations. Ultimately one has to reverse it to see the results on the original representations. Here, the signal is processed by applying the inverse DCT-IV and the unfolding operation in order to generate used features. The inverse DCT-IV is the same as given in Eq. 11.2. For DCT, see Part I.

Inspecting feature transformation, the spectrum of the finite length signal is shown in Eq. 11.3.

$$y_j[n] = \begin{cases} b_j[n]y[n] + b_j[2a_j - n]y_j[2s_j - n] & \text{if } a_j \leq n \leq a_j + r \\ y_j[n] & \text{if } a_j + r \leq n \leq a_{j+1} - r \\ b_j[n]y[n] + b_j[2a_{j+1} - n]y_j[2a_{j+1} - n] & \text{if } a_{j+1} - r \leq n \leq a_{j+1} \end{cases} \quad (11.3)$$

The outputs of this unfolded form $y[n]$ are our speech features used for classification and recognition. The unfolding operator is now shown in Eq. 11.4.

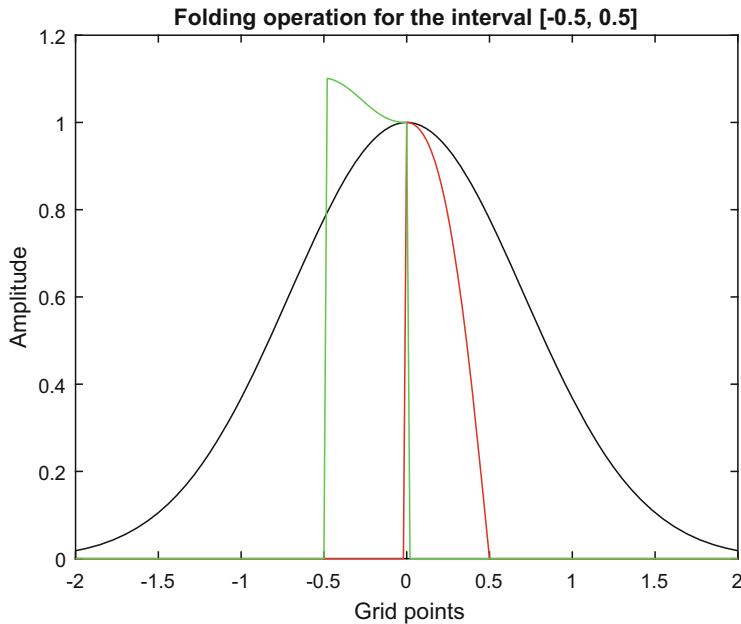


Fig. 11.5 Windowing and folding in wavelet packet analysis

$$s[n] = \begin{cases} b_j[n]s_j[n] - b_j[2a_j - n]s_j[2a_j - n] & \text{if } a_j \leq n \leq a_j + r \\ s_j[n]s_j[2a_j - n] & \text{if } a_j + r \leq n \leq a_{j+1} - r \\ s_j[n + 1] & \text{if } a_{j+1} - r \leq n \leq a_j + 1 \end{cases} \quad (11.4)$$

11.2.1.6 Representation Kinds

i. Parametric Representations

Here the features contain the parameters that are extracted from each frame containing them. The common transformation for this uses the inverse FFT or discrete cosine transforms (DCT). The features are sometimes perceptually adapted by transforming linear frequency into some non-linear frequency by using some scales.

ii. Non-parametric Representations

These are defined as having user parameter free algorithms. FFT based analysis is a typical example of this approach. This is a commonly used tool to begin the recognition tasks. These representations and their consequences will be discussed next.

11.3 Spectral Analysis and Feature Transformation

In the rest of the book, the cepstrum is used several times for spectral analysis. Some use of cepstral analysis is on frequency scales. It is computed by taking the Inverse Fourier Transform (IDFT) of the logarithm of the squared magnitude in the DFT domain.

11.3.1 Parametric Feature Transformations and Cepstrum

The purpose of the cepstrum is to return a frequency response such that the phase of frequency corresponds to a stable and minimum phase system. Cepstrum is computed from taking the Inverse Fourier Transform (IFT) of the logarithm of the estimated spectrum of a signal. The cepstrum parameters for the windowed signal are denoted by $\zeta_m(k)$ and are computed by Eq. 11.5:

$$\zeta_m(k) = \frac{\sum_{n=0}^{N-1} s_m(k) e^{2\frac{\pi}{N}kn}}{N} \quad (11.5)$$

The whole cepstral operation in order to extract cepstrum features is illustrated in Fig. 11.6. In this figure, the signal is first segmented into windowed signals and the discrete Fourier transform (DFT) of the framed signals is computed on the windowed frame. The optional reduction is depicted following the FFT. This data reduction is deployed because:

- Human hearing does not follow linear scale; rather a scale that in literature has adopted its inventor's name, i.e.: "Bark" scale or "Mel" scale.
- Application of reduction scale lowers considerably computational load without degradation on the signal representation, hence it is advisable to perform it. The idea is derived from the fact that two adjacent frequencies can be heard as two tones, e.g., $f_1 = 100$ Hz and $f_2 = 200$ Hz, but not the tones $f_1 = 5100$ Hz and $f_2 = 5200$ Hz which are heard as one tone.

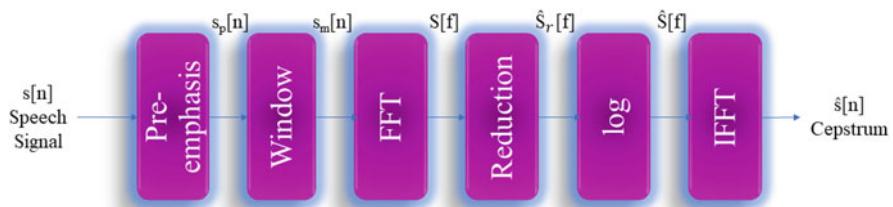


Fig. 11.6 Block diagram of procedure deployed computing cepstrum

Then, an inverse DFT (the efficient implementation of it called FFT, or sometimes an inverse DCT) is taken on the logarithm of the DFT/FFT transformed signal. The extracted features do not show how good the structure of the features is. By good we mean the ability to differentiate between features and whether they are treated well by the cepstrum feature extraction technique. “Treated well” is user defined and not universal.

11.3.2 Standard Feature Extraction Techniques

Below we provide some commonly used feature extraction techniques that also incorporate elements of the applications. Examples are for instance in the speech application of human hearing at several stages and the biomedical applications (see Part III). The general aspects provided here refer to arbitrary signals. There are different measurements and distribution properties to be used as features. We provide first a list of conventional feature parameters. The extraction methods are not recommended but are simply described.

11.3.3 Frame Energy

A quantitative measure for frequency responses of the output of a system in responding to an input is used to characterize the dynamics of the system. It measures mainly the magnitude of the output. If the system is time-invariant, then the frequency response will also not vary with time. FFT is a standard approach to measure the magnitude frequency response of each frame. Frame energy is the measure of the energy of a frame in Eq. 11.6. Fast Fourier Transforms (FFT) based spectral analysis reveals the energy distribution in the frames in terms of frequency. The magnitude measure of the Fourier Transform $s[n]$ is computed as

$$\sqrt{\sum_{n=0}^{N-1} \frac{s^2[n]}{N}} \quad (11.6)$$

where N is the length of the frame.

11.3.3.1 Spectral Envelope Again

A spectral envelope is a piece-wise spectral information. It is used parametrically and also non-parametrically using FFT to characterize the signal. A spectral envelope is derived either by a Fourier transformed windowing method in a non-parametric way or in a linear prediction model based parametric way. In the

non-parametric method, the windowing or low pass filter based log magnitude spectrum is computed to extract the piece-wise spectral information of the system. In the parametric method, the amplitude response of the all-pole filter is analyzed to obtain the piece-wise spectral information of the system. The following list shows the major influence factors.

Log Energy These are logarithmic computations of the short term energy of the speech signal.

Delta Cepstrum These are the derivatives of the cepstrum features. They are used to capture the dynamic behavior or the underlying information of the speech process by taking the derivatives of the primarily extracted feature parameters. Particular examples are the derivatives of the energy and the velocity and the acceleration of the features as an indication to get a realization of the time variation of the signal.

Spectrogram The spectrogram is a graphical representation of the energy density as a function of the frequency. Spectrograms of the speech signals often analyze the phonemes and their transitions. In a linguistics sense, phoneme can be defined as some phonetically distinct articulations.

Entropy Entropy characterizes the behavior of the random variables. It is quite often useful to estimate the probabilities of events to find the hypothesis of the smallest error.

Next we give a short description of the prediction components and some methods connected with them.

11.4 Linear Prediction Coefficients (LPC)

Linear prediction is a special method that is not only useful for feature extraction. It is rather a general application method in its own right. The use of linear models for approximating results is in general a widespread method in mathematics. In the sequel we will discuss this several times. The basic idea of this technique in this context is first to obtain the auditory spectrum and then to obtain the features. In the [Linear Predictive Coding Coefficients](#) (LPC) feature extraction the signal is segmented and windowed by using Eq. 11.7.

$$s_m[n] = s[n] \times w[n] \quad (11.7)$$

Thus the framed or the windowed signal $s_m[n]$ is the multiplication of the signal $s[n]$ by the weighted window function $w[n]$; where m is the applied weight. When multiplying the signal $s[n]$ by the window function $w[n]$ we attain the signal frames denoted by $s_m[n]$ for $m = 1, 2, \dots, M$ and $n = 0, 1, \dots, N - 1$. These are the shift samples between adjacent segments. This shifting is sometimes known as overlapping. Mostly, it can take place at each one half or two thirds of the length of the signal where the length of the signal is assumed to be the same as the length of

the signal block. A widely used type of window is the Hamming window function. However, there are many different types of window functions available: Blackman, Chebyshev, Hamming, Hanning, Kaiser just to name a few. Some commonly used window functions are the rectangular window functions. One defines it as given in Eq. 11.8:

$$w[n] = 1 \quad \text{for } 0 \leq n \leq N - 1 \quad (11.8)$$

The Hanning window function is presented in Eq. 11.9 and in Fig. 11.7.

$$w[n] = 0.5 + 0.5 \cos\left(\frac{2\pi n}{N-1}\right) \quad \text{for } 0 \leq n \leq N-1 \quad (11.9)$$

The Hamming window is presented in Eq. 11.10 and Fig. 11.8.

$$w[n] = 0.54 + 0.36 \cos\left(\frac{2\pi n}{N-1}\right) \quad \text{for } 0 \leq n \leq N-1 \quad (11.10)$$

One purpose of the windowing is to improve the leakage problem and results in a spectral bias in the frequency analysis. The leakage problem arises when a signal is truncated into blocks.

This leads to the autocorrelation. It is the cross-correlation of a signal with itself, hence the word auto(-correlation). Informally it is the similarity between observations as a function of the time lag between them. Sometimes the term is used interchangeably with autocovariance. That will further be discussed below.

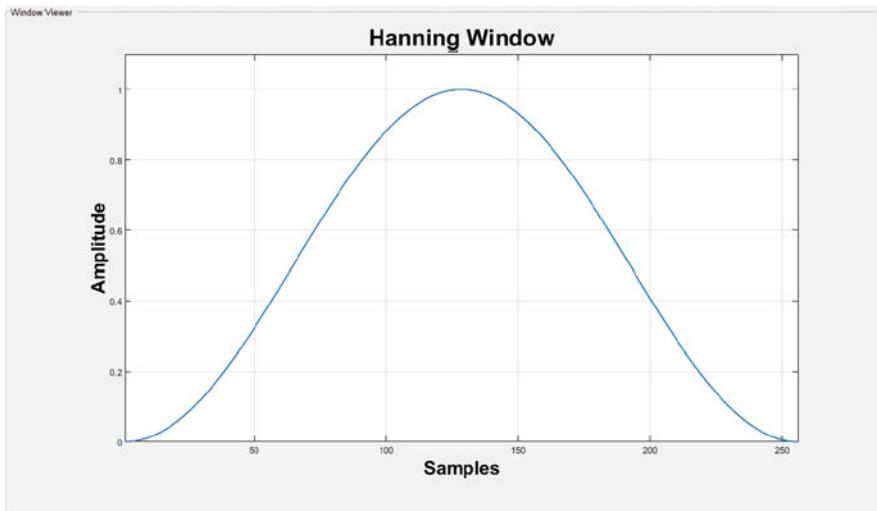


Fig. 11.7 Example of 256 tap (points) Hanning window

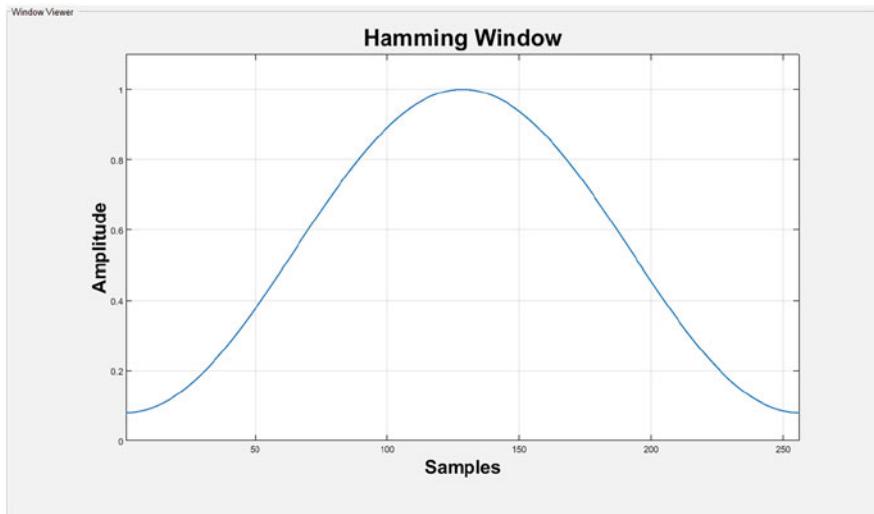


Fig. 11.8 Example of 256 tap (points) Hamming window

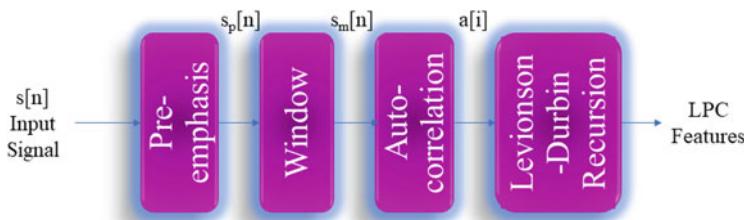


Fig. 11.9 Block diagram depicting computation of LPC features

First we turn to an example of feature generation. Figure 11.9 shows an example of the feature extraction. The LPC computation illustrates how different techniques are used. This includes autocorrelation; the technique used to generate LPC features. Further information involving technical details from the perspective a the user of library functions is presented next:

- First, the file format of the input has to be specified. This can be HTK, WAV, NIST or any other user format.
- WAVEFORM is the default file type that specifies the input data kind.
- TARGETFORMAT declares the desired format of the output. Only waveform data can be the output. HTK is the default input format.
- TARGETKIND specifies the output that specifies kind of data is produced. For example typical speech features produced are of the form of mel filter cepstral coefficients—MFCC for short, of speech that may includes its derivatives.

- The other useful parameters of selection options are PLP cepstral coefficients, MELSPEC.
- WINDOWSIZE specifies the length of the window in 100ns units.
- NUMCHANS specifies the number of mel-frequency filter banks to use.
- NUMCEPS specifies the number of cepstra to keep. The total acoustic feature vector has dimension 12 cepstra in addition to the energy of the frame of speech.
- LOFREQ specifies the lower cut-offs frequency of the first filter in the filter bank analysis.
- HIFREQ specifies the upper cut-offs frequency of the last filter in the filter bank analysis.
- CEPLIFTER specifies that the cepstrum should be lifted to deemphasize the lower order cepstra. Lifting is useful because the low order cepstra are usually much larger in amplitude than the high order cepstra. Without lifting, speech recognition distance measured would completely ignore the high-order cepstra.

11.5 Linear Prediction Cepstral Coefficients (LPCC)

First we discuss an extended version of LPC, namely LPCC. The LPCC feature extraction technique is a combination of the LPC analysis and the cepstrum analysis discussed in Sect. 11.3.1. The basic idea of this technique for LPCC is the same as in the LPC analysis. In this technique the windowed signal is used for computing the autocorrelation and the power spectrum and the inverse DFT of the logarithm of the power spectrum is computed for the LPCC features. A block diagram of the LPCC feature extraction technique is shown in figure below (Fig. 11.10).

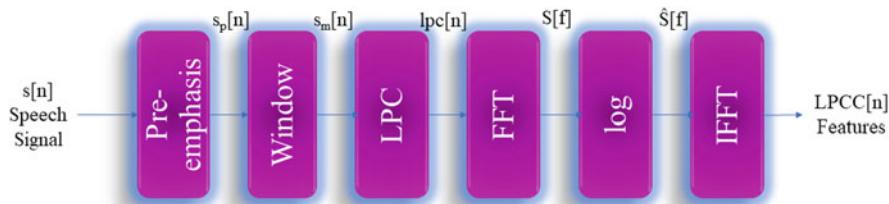


Fig. 11.10 LPCC feature extraction

11.6 Adaptive Perceptual Local Trigonometric Transformation (APLTT)

For completeness we discuss the APLTT feature extraction next. The steps involved with Adaptive Perceptual Local Trigonometric Transformation (APLTT) are outlined below:

- Spectral Shaping: The signal is decomposed into blocks using a lapped transformation followed by a folding operator.
- Spectral Analysis: This is done by computing discrete cosine transform IV (DCT—IV).
- Perceptual Mapping: This is done by computing the perceptual spectral information using a Bark scale. This is an extension of the shift invariant local trigonometric transformation (SILTT).
- Perceptual Feature Transformation: The perceptual features are the perceptual entropy (PE). This is a modification of the SILTT.
- Parametric Feature Transformation: This is done by computing the inverse of the discrete cosine transform IV (IDCT—IV) followed by an unfolding operation.

11.7 Search

There is an old slogan saying that humans spend most of their time for sleeping and searching. Search is no fun in itself but done only for successful results of the activity. In fact, search can be annoying. Therefore one wants to do it as short and efficient as possible. For this search needs a plan. In this respect search has something in common with prediction. Therefore search is essential for prediction. This happens whenever one has not only one way to proceed but a branching occurs. If a search does not seem successful backtracking to a branching point takes place. Search has a goal that one wants to achieve. This goal comes up when one has to make decisions that branch over and over and the result arises at the end of the branches. Then one has to search for the best result. Search enters our discussion at several points in particular with respect to machine learning. Search in particular plays a big role in machine learning.

11.7.1 General Search Model

We present here a formal model for search.

Definition

- i. A search model consists of a tuple (S, T, G, C) . For a set S (called states) and a set $T \subseteq S \times S$ (called transitions) the pair (S, T) is called a basic *search*

model. A search instance is a pair (s, G) where $s \in S$ (called the initial state) and $G : S \in \{yes, no\}$ (called the termination criterion, or goal test). Going from one state to another one is along a transition is called a search step.

A control function C is a mapping considering the search steps. It selects the next steps: with $C(s) \in \{s' | (s, s') \in T\}$.

- ii. If a state $\{yes\}$ is reached then the search stops successfully.
- iii. This defines the possible results when the transition T is called.
- iv. A control function needs not always to exist. If it exists the next search state depends only on the last state. The triple (S, T, C) is often called an autonomous search process.

Important examples of search models are A* and Beam search.

Now we assume that there is a definition of “best”. With the special case of Viterbi decoding the search goal is to find best state sequence along a single path at certain time t , the best state and the best score. This occurs if one wants to find a particular process. Viterbi decoding in Fig. 11.11 shows the matching process, e.g., an example pattern with the given model. Given model in this figure is representing an acoustic model. Viterbi decoding is used for many applications especially in information and communications. Machine learning is a wide area. We just mentioned concepts of learning where some specific aspects are of interest for handling signal processes. The general goal is to extract properties from given knowledge (mostly in the form of examples). Very popular is unsupervised learning; this procedure is natural for stochastic processes. For general information on machine learning and Gaussian processes we recommend the popular book (Rasmussen et al. 2006). Search plays a major role in predictions. There one often has the choice between different alternatives and selecting the right one is a search problem.

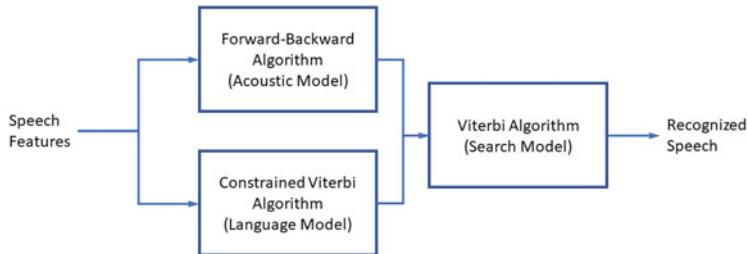


Fig. 11.11 Viterbi decoding in speech pattern recognition

11.8 Predictions

We mentioned predictions several times. Now we discuss specific methods.

11.8.1 Purpose

Predictions say something about an unknown event that is often in the future. For processes this means that one wants to find out what the next process step, or steps, could be or what the long range properties could be. In stochastic processes we have no logical rules and we have to rely on estimates and probabilities. Hence the result is uncertain and in principle every object is a candidate to be the predicted one. If probabilities are present one generally chooses the element with the highest probability. This is problematic for hidden probabilities. In order to determine the element with the highest probability one is confronted with an optimization problem. It is of particular difficulty if the states are not observable or only partially observable. For HSM the questions are now:

- i. On which basis do we get the probabilities?
- ii. Which knowledge can one use?
- iii. What is supporting for the prediction?
- iv. Which optimization technique should be applied?

The information about probabilities involved is obtained from studying (many) examples and applying machine learning techniques as necessary. In addition there may be other principles supporting the search. One such principle is that events are not completely independent. Another principle is that many of the events are redundant. The problem here is to replace the many events by fewer ones without losing essential information. This process is exploited in feature extraction.

A basic problem for optimization is “what is better?” This is related to utilities, as discussed above, where better means more useful. In the linear prediction (LP) approach the signal process is assumed to be an auto-regressive process. In this process type, one uses previous process results for further reasoning. It relies on the tacit assumption that the values are not completely independent of each other, a situation that often occurs in practice.

11.8.2 Linear Prediction

A particular auto-regressive method is linear prediction. There the parameters of the process are approximated by applying linear prediction (LP) analysis. This is done by trying to approximate the predicted value in terms of linear functions of earlier values. Therefore the coefficients of those linear functions have to be determined. LP

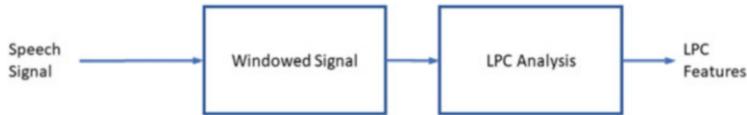


Fig. 11.12 Linear prediction analysis

Visualization in Fig. 11.12 provides a first block diagram: Suppose the current signal is $s[n]$ and we want to estimate s by some \hat{s} . In LP analysis, the current sample $s[n]$ is predicted by using linear combinations of the past p -samples producing a value α_i , as indicated in Eq. 11.11:

$$\hat{s}[n] = \sum_{i=1}^p \alpha_i s[n - i] \quad (11.11)$$

An expanded form of Eq. 11.11 is given in Eq. 11.12:

$$\hat{s}[n] = \alpha_1 s[n - 1] + \alpha_2 s[n - 2] + \cdots + \alpha_p s[n - p] \quad (11.12)$$

The prediction \hat{s} of s may not be accurate since the assumption of linearity may not be true. The goal is to minimize the error in general. A simple way would be to store the error $e[n]$ from Eq. 11.13.

$$e[n] = s[n] - \hat{s}[n] \quad (11.13)$$

This works only if the true value is known. For the unobservable stochastic situation a more involved concept is needed. In a stochastic approach one uses the expected value E of the squared values as prediction error. The definition is shown in Eq. 11.14.

$$E\{|e|^2\} = \frac{1}{N} \left\{ \sum_{n=0}^{N-1} \left(s[n] - \sum_{i=1}^p \alpha_i s[n - i] \right)^2 \right\} \quad (11.14)$$

Terms $\alpha_i s$ remain the same in Eq. 11.14. Equation 11.14 is more accurate. Each sample in the segment s is estimated by a linear combination of the α_i for $i = 1, 2, \dots, p$. Now we extend the error term to the average squared value. The mean squared error term is used here following the ergodicity concept of the random sequences. The term “Ergodic” means here that in the long run the values are not determined by the initial ones. Therefore any additional incoming information is useful.

11.8.3 Mean Squared Error Minimization

The task and the general method is to minimize the expected value of the squared values of the prediction error. When the Euclidean distance describes a distribution or the squared Euclidean errors are considered, then the underlying concept of the process is presumably a Gaussian distribution. The mean of the discrete random process is obtained by the average summation of the ensemble. To find the minimum mean squared error one follows a standard procedure. One needs to differentiate $E^2(e)$ with respect to α_k for $k \in Z$ for which E becomes minimal. Assuming quadratic surface, it is computed according to Eq. 11.15:

$$E = \frac{1}{N} \sum_{n=0}^{N-1} (s[n] - \hat{s}[n])^2 = \frac{1}{N} \sum_{n=0}^{N-1} e^2[n] \quad (11.15)$$

If the order of predictor is picked to be p , then there will be p derivatives when $E(e^2)$ is differentiated.

The derivation is computed by its derivatives with respect to coefficients in Eqs. 11.16 and 11.17.

$$\eta'_k = \frac{\delta E}{\delta \alpha_k} = \frac{\delta}{\delta \alpha_k} \frac{1}{N} \sum_{n=0}^{N-1} (s[n] - \hat{s}[n])^2 \quad (11.16)$$

$$\begin{aligned} \frac{\delta E}{\delta \alpha_k} &= \frac{1}{N} \frac{\delta}{\delta \alpha_k} \sum_{n=0}^{N-1} \left(s[n] - \hat{s}[n] \right)^2 = \frac{1}{N} \frac{\delta}{\delta \alpha_k} \sum_{n=0}^{N-1} \left(s[n] - \sum_{i=1}^p \alpha_i s[n-i] \right)^2 \\ \frac{\delta E}{\delta \alpha_k} &= \frac{1}{N} \sum_{n=0}^{N-1} \frac{\delta}{\delta \alpha_k} \left(s[n] - \sum_{i=1}^p \alpha_i s[n-i] \right)^2 \\ \frac{\delta E}{\delta \alpha_k} &= \frac{1}{N} \sum_{n=0}^{N-1} 2 \left(s[n] - \sum_{i=1}^p \alpha_i s[n-i] \right) \left(-\frac{\delta}{\delta \alpha_k} \sum_{i=1}^p \alpha_i s[n-i] \right) \\ \frac{\delta E}{\delta \alpha_k} &= \frac{1}{N} \sum_{n=0}^{N-1} 2 \left(s[n] - \sum_{i=1}^p \alpha_i s[n-i] \right) (-s[n-k]) \end{aligned} \quad (11.17)$$

Assuming that error surface is quadratic, one will equate the first derivatives with respect to the parameter α_k to zero as shown in Eq. 11.18; that is, we are determining $\{\alpha_k\}$ for which error E is minimal:

$$\eta'_k = \frac{\delta E}{\delta \alpha_k} = 0 \quad (11.18)$$

This yields to the p equations written in Eq. 11.19 for $i = 1, \dots, p$,

$$2 \sum_{n=0}^{N-1} \left(-s[n]s[n-k] + \sum_{i=1}^p \alpha_i s[n-i]s[n-k] \right) = 0 \quad (11.19)$$

finally yielding to the Eq. 11.20, below.

$$\frac{\delta \hat{s}[n]}{\delta \alpha_k} = s[n-k] \quad (11.20)$$

Please note that the derivative on the left side of Eq. 11.20 leads to Eq. 11.21 where k is a dummy variable. In this equation, we see, we need to find out i many past α coefficients to obtain the current sample $\forall i$ i.e., $i = 1, 2, \dots, p$. In order to obtain the current estimate for the sample, we need to obtain past samples from the prediction window in order to perform the estimate as we can see in Eq. 11.21. Predictions, namely forward, backward and their combination can be applied for this. Finding the best approximation is an optimization problem that we will now discuss.

$$\sum_{n=0}^{N-1} s[n]s[n-k] = \sum_{k=1}^p \alpha_k \sum_{n=0}^{N-1} s[n-i]s[n-k] \quad 1 \leq i \leq p \quad (11.21)$$

11.8.4 Computation of Probability of an Observation Sequence

The probability of a observation sequence, $O = o_1, o_2, \dots, o_T$ given the model λ can be computed by a thorough enumeration of every possible state sequence of length T (the number of observations). For instance, tN states based model, $\mathbf{Q} = \{q_1, q_2, \dots, q_N\}$ has N^T possible state transition sequences as shown in Eq. 11.22.

$$P(\mathbf{O}|\lambda) = \sum_{all \mathbf{Q}} P(\mathbf{O}, \mathbf{Q}|\lambda) \quad (11.22)$$

which is detailed in Eq. 11.23.

$$P(\mathbf{O}, \mathbf{Q}|\lambda) = P(\mathbf{O}|\mathbf{Q}, \lambda) P(\mathbf{Q}|\lambda) \quad (11.23)$$

For a length T fixed state sequence $\mathbf{Q} = q_1, q_2, \dots, q_T$, the probability of the observation sequence \mathbf{O} given the state sequence, assuming statistical independence of observations can be formulated by Eqs. 11.24 and 11.25

$$P(\mathbf{O}|\mathbf{Q}, \lambda) = \prod_{t=1}^T P(\mathbf{o}_t|q_t, \lambda) \quad (11.24)$$

and

$$P(\mathbf{O}|\mathbf{Q}, \lambda) = b_{q_1}(\mathbf{o}_1)b_{q_2}(\mathbf{o}_2) \cdots b_{q_T}(\mathbf{o}_T) \quad (11.25)$$

where $b_{q_i}(o_i)$ is the conditional probability $P(o_i|q_i)$. The probability of such a state sequence q can be expressed by Eq. 11.26.

$$P(\mathbf{Q}, \lambda) = \pi_{q_1}b_{q_1q_2}b_{q_2q_3} \cdots b_{q_{T-1}q_T} \quad (11.26)$$

where $b_{q_i, q_{i+1}}$ is the conditional probability $P(q_{i+1}|q_i, \lambda)$.

The joint probability of \mathbf{O} and \mathbf{Q} , i.e., the probability that \mathbf{O} and \mathbf{Q} occur simultaneously, is simply the product of the previous terms:

$$P(\mathbf{O}, \mathbf{Q}|\lambda) = P(\mathbf{O}|\mathbf{Q}, \lambda)P(\mathbf{Q}|\lambda) \quad (11.27)$$

The probability of \mathbf{O} given the model λ is obtained by summing this joint probability over all possible state sequences \mathbf{Q} :

$$P(\mathbf{O}|\lambda) = \sum_{\forall \mathbf{Q}} P(\mathbf{O}|\mathbf{Q}, \lambda)P(\mathbf{Q}|\lambda) \quad (11.28)$$

Expanded, that equation becomes:

$$P(\mathbf{O}|\lambda) = \sum_{q_1, q_2, \dots, q_T} \pi_{q_1}b_{q_1q_2}b_{q_2q_3} \cdots b_{q_{T-1}q_T} \quad (11.29)$$

Interpretation of the previous expression: Initially at time $t = 1$ we are in state q_1 with probability π_{q_1} , and generate (or accept) the symbol o_1 (in this state) with probability $b_{q_1}(o_1)$. In the next time instance $t = t + 1$ ($t = 2$) transition is made to state q_2 from state q_1 with probability $b_{q_1q_2}$ and generate (or accept) the symbol o_2 with probability $b_{q_2}(o_2)$.

The process is repeated until the last transition is made at time T from state q_T from state q_{T-1} with probability $b_{q_{T-1}q_T}$ and generate the symbol o_T with probability $b_{q_T}(o_T)$.

Practical Problem

Calculation requires $\approx 2T \cdot N^T$ of multiples (since there are N^T such sequences). For example: $N = 5$ (states), $T = 100$ (observations) $\Rightarrow 2 \times 100 \times 5^{100} = 10^{72}$ computations! Hence, a more efficient procedure is required. That procedure is known as the **Forward** and **Backward** algorithm.

11.8.5 Forward and Backward Prediction

There are more details about forward and backward prediction for applications in Part III, however the fundamentals are presented here. Suppose we have a stochastic process $s[n]$. If we look at any time t , two time segments can be identified:

- i. Past where $\{n, n < t\}$ the time points before present time t .
- ii. Future where $\{n, n > t\}$ the time points after present time t .

If one has a time dependent sequence of n points then one can freely choose some t and gets the splitting of the time points. If one takes some p one can estimate $s[n]$ in two ways:

- i. Using the p time points before n : *Forward prediction (FLP)*
- ii. Using the p time points after n : *Backward prediction (BLP)*.

For both estimates we can take any appropriate method, for instance the linear prediction. If one has the choice of t then this amounts to the choice of the methods. In terms of the prediction equations one gets the following analysis. The linear prediction we provided so far is mainly the forward linear prediction. The current value can also be predicted from the future values and this is backward prediction. The backward prediction can be derived efficiently from the forward prediction by delaying the forward prediction by only one sample. This requires that all events have to be known.

We repeat both types of predictions. Equation 11.30 shows forward linear prediction (FLP). Equation 11.31 shows backward linear prediction with an acronym (BLP). This predicts p many additional future samples the arguments used in the equations varied.

$$s[n] = \sum_{i=1}^p a_i s[n-i] + g_s u_s[n] \quad (\text{FLP}) \quad (11.30)$$

$$s[n] = \sum_{i=1}^p a_i s[n+p-i] + g_s u_s[n] \quad (\text{BLP}) \quad (11.31)$$

Both equations, Eqs. 11.30 and 11.31, give rise to recursions if one applies the prediction steps iteratively. In the sequel, we describe three steps such as initialization, induction, and termination to describe recursion algorithms (as indicated in the introduction). The two equations use in summary the same arguments. Therefore one cannot expect that they produce different results.

11.8.5.1 Forward Prediction

In a given model the probability of the set of observations in a specific state sequence has to be estimated. Given the model, the probability of o_t being at state i at time t can be estimated by the forward probability denoted by α . The forward probability can in principle be computed in the recursion by using Eq. 11.32.

$$\alpha_t(i) = p(o, q|\lambda) = p(o_1, o_2, \dots, o_i, q_t = i|\lambda) \quad (11.32)$$

The forward algorithm estimates the likelihood $p(o|\lambda)$ (which means for the given model the probability of the observation at a certain time and at certain state) by the following recursion:

Initialization

This uses the initial parameters to start the evaluation and is shown in Eq. 11.33 for $t = 2, 3, \dots, T$ and $j = 2, 3, \dots, N$. This provides the estimate of the likelihood:

$$\alpha_1(i) = \pi_i b_i(o_1) \quad \text{for } 1 \leq i \leq N \quad (11.33)$$

Induction

The probabilities of the observations from the past to the present state are computed using the previous probabilities, transition probabilities and observation probabilities in order to estimate the likelihood of the observations. The computation of the observation for a state is modeled by Eq. 11.34 where T is the length of the each features observations:

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t) \quad \text{for } 1 \leq T \text{ and } 2 \leq j \leq N \quad (11.34)$$

Termination

This gives the estimate of the likelihood of the observation for a state given the model by Eq. 11.35 where T is the length of the each features observation:

$$\alpha_T(n) = p(o|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (11.35)$$

11.8.5.2 Backward Prediction

The backward search is done by the backward algorithm. The backward probability β denotes the probability of the observations o_T through o_{t+1} being in state i at time t given a HSM model λ by Eq. 11.36. Here the probability of the future sequence conditioned on the present state j at time t is computed.

$$\beta_t(i) = p(o, q|\lambda) = p(o_{t+1}, o_{t+2}, \dots, o_T, q_t = i|\lambda) \quad (11.36)$$

Initialization

The model in state i at time T is 1. The transition of the observation is finished at time $T + 1$ and this is also:

$$\beta_T(i) = 1 \quad \text{for } 1 \leq i \leq N \quad (11.37)$$

Induction

This step computes the likelihood of the observation given the model. It is shown in Eq. 11.38

$$\beta_t(i) = \sum_{j=1}^N b_j(o_{t+1}) a_{ij} \beta_{t+1}(j) \quad \text{for } 1 \leq t \leq T \text{ and } 1 \leq j \leq N \quad (11.38)$$

Termination

One also needs to give end of the backward procedure. Suppose β_0 is the state at the beginning signal that emits the π values in the transition to the first states at time 1. Then we get Eq. 11.39.

$$\beta_1(1) = \sum_{j=1}^N \pi_j b_j(o_1) \beta_1(j) \quad (11.39)$$

11.8.6 Forward-Backward Prediction

This approach combines the prediction kinds just introduced. The point is here that one does not get essentially new predictions. The expectation is rather to obtain a higher efficiency in the prediction process. For the combination one needs many examples. A particular (and recommended) method is the Baum-Welch algorithm.

11.8.6.1 Baum-Welch Algorithm

This algorithm is used for training model. The Baum-Welch technique uses the forward-backward algorithm to estimate model parameters from the observation sequences. The general idea is expressed in Eq. 11.40.

$$P(s_i|s_j) = \frac{\text{Expected number of transition from state } s_j \text{ to state } s_i}{\text{Expected number of transition out of state } s_j} \quad (11.40)$$

Additional required details are provided below: The combination of the forward and backward probability is given in Eq. 11.41.

$$p(o, q_k = i | \lambda) = \alpha_t(i) \beta_t(i) \quad (11.41)$$

This expression is used to find the model parameters from the forward and backward direction in the frame of Estimation Maximisation (EM) algorithm. The Baum-Welch technique uses the following steps for training the model and the model parameters of the observation sequences:

- (1). Compute the forward probabilities α by using the forward algorithm.
- (2). Compute the backward probabilities β by using the backward algorithm.
- (3). Compute the transition probabilities A and the emission probabilities B at the current state using the observation sequences.
- (4). Compute the new model parameters μ , σ and c .
- (5). Compute the new log likelihood of the model.
- (6). Stop computations when there is no change in the log-likelihood.

The method is applied to the Gaussian Mixture Model (GMM) model. For the GMM model parameters estimations, the solutions for the re-estimation formula for $hatc$, $\hat{\mu}$, and $\hat{\Sigma}$ are estimated for the observations $b_j(o_t)$.

Initialization

It starts with the probability of the model given at a certain state i at time t . It is shown by Eq. 11.42 given below.

$$p(o, q_k = i | \lambda) = \alpha_t(i) \beta_t(i) \quad (11.42)$$

Induction

Apply the forward and the backward methods to Eq. 11.42.

Termination

It stops learning when the likelihood is the same for both the forward and backward algorithms. The Baum-Welch algorithm works for any stochastic process. Technically very close to it is the Viterbi algorithm that is an approximate method to Baum-Welch for computing likelihoods. Originally it was developed for speech recognition but it is of more general character and so it is presented here.

11.8.6.2 Viterbi Algorithm

In the Viterbi search, the best state sequence along a single path at a certain time t and the best score are computed. The highest likelihood $o_t(i)$ in state i at time t is given in Eq. 11.29 or 11.43.

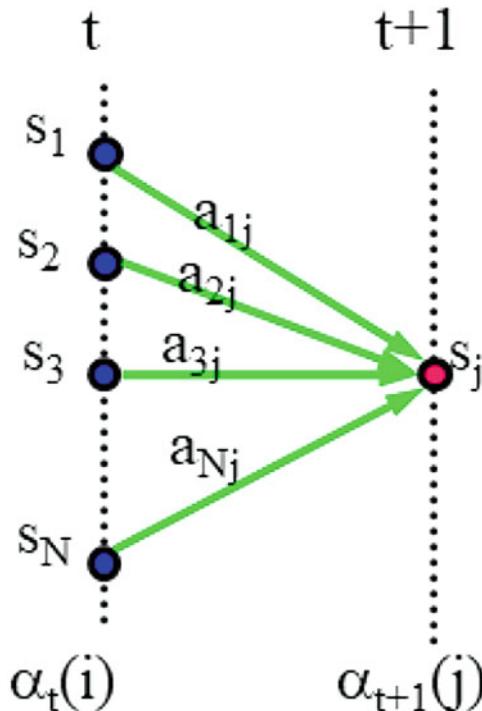


Fig. 11.13 Viterbi Algorithm as applied to the network of the type shown

Notation

The $\Psi_t(j)$ is the best state prior to state j at time t . a_{ij} denotes the transition probability from i to j in the transition matrix A . It is stated in Eq. 11.43.

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} p(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t | \lambda) \quad (11.43)$$

$$\Psi_t(i) = \operatorname{argmax}_{q_1, q_2, \dots, q_{t-1}} p(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t | \lambda) \quad (11.44)$$

Figure 11.13 represents the processes described in Eqs. 11.43 and 11.44. The Viterbi algorithm computes the optimal state sequence q_1, q_2, \dots, q_{T-1} related to the observations with respect to their joint probability $O = o_1, o_2, \dots, o_N$ given the model. Steps involved in applying Viterbi algorithm.

Initialization

Equation 11.45 given below denotes the transition that starts from the initial state π and it ends up at the state b_i for $1 \leq i \leq N$ and the observation o at time $t = 1$. We conclude with Eq. 11.45

$$\delta_t(i) = \pi_i b_i o_1 \quad \text{for } 1 \leq i \leq N \quad (11.45)$$

Induction

In the recursion one finds the path that evaluates to a maximum likelihood considering the best likelihood path with one step shorter and the transition from it. This is then multiplied by the current likelihood given the current state and thus the best path is obtained through the induction in Eq. 11.46:

$$\begin{aligned}\delta_t(i) &= \max_{1 \leq j \leq N} \delta_{t-1}(j) a_{ij} b_j(o_t) \quad \text{for } 1 \leq i, j \leq N \\ \Psi_t(j) &= \arg \max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij} \quad \text{for } 1 \leq i, j \leq N\end{aligned}\quad (11.46)$$

Termination

The search is terminated when the end of the observation sequence is reached at a given final state. As the result of applying this algorithm, the best likelihood is computed.

The algorithm is computed in the log domain, to avoid underflow errors. Any state can be denoted as a valid end-of-utterance state. The maximization occurs only over those states which are used in the problem as states for the valid end-of-utterance states.

Figure 11.14 illustrates inner workings of the Viterbi algorithm for language purposes.

Relationships Between Viterbi search and Dynamic Time Warping (DTW)

DTW is closely related to the Viterbi method. Essentially they have the same goal. We provide a short overview over commonalities and differences of the two.

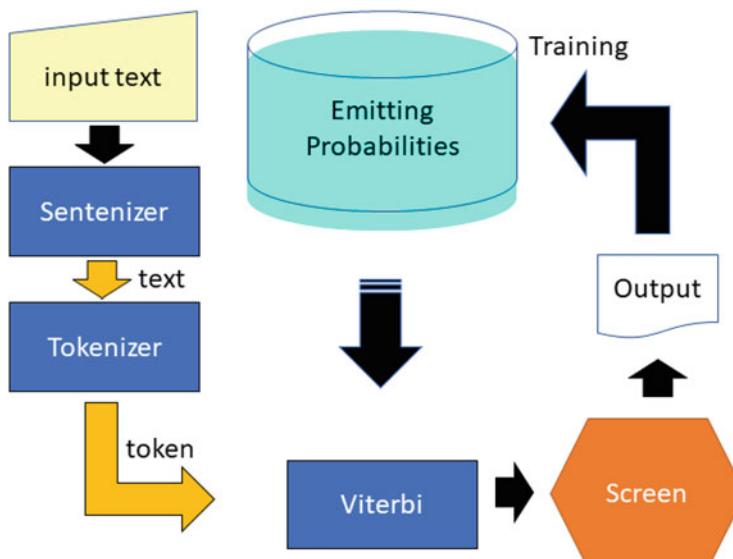


Fig. 11.14 Viterbi Algorithm as applied to text

- Both look for a best path.
- For both the best path utterance is computed from the best path when time $t = T$.
- DTW uses only previous data points.
- Viterbi search uses previous time values.
- DTW cost D for a point (x, y) is computed using cumulative costs for previous points, transition costs (path weights), and local costs for current point (x, y) .
- Viterbi probability δ for a time t and state j is computed using cumulative probability for previous time points and states, transition probabilities, and the local observation probability for the current time point and state.

Of course, both approaches can be combined in various ways. One cannot say that one approach is in general better than the other one. If one is interested in costs, DTW may be preferred, otherwise Viterbi is broader. This depends on the user and its utility. In the next chapter there is a description of the methods for classification, using much of the material presented here.

11.9 Background Information

Past

In the 1940s Norbert Wiener developed a mathematical theory for calculating the best filters and predictors for detecting signals hidden in noise, Wiener (1949). Extrapolation, Interpolation, and Smoothing of Stationary Time Series. Proposed novelty was in the solution as a filter used to produce an estimate of a desired or target random process by linear time-invariant filtering of an observed noisy process. The assumptions are stationary signals and there is a noise spectrum.

The Wiener filter minimizes the mean square error between the estimated random process and the desired process. Later approaches refine the Wiener filtering. A successful use was in image analysis. More recent methods can be found in Hamilton (1994).

The technical first step is always to detect the features. The features are smaller than the original signal processes and contain therefore less information. This forces the user to select the right extraction that preserves the information of interest. An example would be Mel Frequency Cepstral Coefficients that lead to a compact representation of the frequency spectrum. Feature extraction is the main technique to solve the problem of getting shorter representations. It starts from an initial set of measured data and generates features. These are derived values with the intention to compact information and present it as real valued vectors.

There are different kinds of features depending on which information one is looking, as discussed in Part III of this book. Because the feature vectors are much shorter than the original signals they will necessarily lose some information. The task of the feature extraction is to preserved as much as possible the information of interest in spite of the noise. This has been achieved in many cases such as LPCC.

In LPCC the input signal is first pre-emphasized using a first order high pass filter. To clarify, there are two things denoted by LPCC:

- (1). Linear Program with Linear Complementarity Constraints; and
- (2). Linear Predictive Cepstral Coefficients.

The APLTT was introduced in the system DANSR, see Paul and Richter (2013). Many data analysis software packages provide feature extraction and dimension reduction methods. Common numerical programming environments such as MATLAB give some of the simpler feature extraction techniques.

Information about the forward-backward approach can be found in Russell and Norvig (2010) and also discussed in Part III of this book. The Baum-Welch algorithm for forward and backward search is named after Leonard E. Baum and Lloyd R. Welch. It is an algorithm that maximizes expectations. It is built on the idea of the forward-backward method. More on the technical details of this algorithm is in Bilmes (1998) and Fazzoli (2013).

The cepstrum was introduced in Bogert et al. (1963). The cepstrum of a signals is a kind of spectrum of the logarithmical frequency. The name is derived from the word spectrum by reversing the first letters (Bogert et al. 1963). The Viterbi algorithm is named after Viterbi (1967). The Viterbi path deals with HSM in order to find the hidden process. It is the most likely sequence of hidden states. The Viterbi algorithm is an algorithm for finding it. For the Viterbi algorithm see Forney (1967).

Suggested

One can find more on the comparison between Viterbi and DTW search and general aspects of HSM in Hosom (2011). It is now also commonly used in speech recognition and speaker diarization (i.e. of partitioning an input audio stream into homogeneous segments according to the speaker identity), computational linguistics, and bioinformatics. For text recognition see also Shinghal and Toussaint (1979). Forward and backward search was introduced and made popular in speech recognition, see again Hosom (2011). For speech and other applications it is of advantage for efficiency. More on recognizers with many data are found in Young (2006). We discuss this in Part III in detail.

11.10 Exercises

Exercise 1 Write a C++ program to compute LPCC features.

Exercise 2 What is the frame energy if $s[n]$ is constantly increasing by a number d .

References

- [Wiener1949] Wiener, N. (1949). Extrapolation, Interpolation, and Smoothing of Stationary Time Series. New York, Wiley.
- [Hamilton1994] Hamilton, J.D. (1994) Time Series Analysis. Princeton University Press, New Jersey, USA.
- [Russel2010] Russell R., Norvig P. (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Pearson Education Prentice-Hall.
- [Paul2013] Paul, S., Richter, M. M. (2013). A dynamic automatic noisy speech recognition (dansr) system for a single channel hybrid noisy industrial environment, Vol. 21. ICA, June 2013.
- [Bogert1963] Bogert B.M., Healy M.J.R., Tukey J.W. (1963): The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, and Shape Cracking. Proceedings of the Symposium on Time Series Analysis (ed. M. Rosenblatt).
- [Vitebi1967] Viterbi, A. (1967): Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In: IEEE Transactions on Information Theory. 13, Nr. 2.
- [Forney1967] Forney Jr., G. D. (1967): The Viterbi Algorithm. In: In Proceedings of the IEEE. 61, Nr. 3, 1973
- [Frazzoli2013] Frazzoli, E. (2013). Intro to Hidden Markov Models the Baum-Welch Algorithm. Aeronautics and Astronautics, Massachusetts Institute of Technology.
- [Hosom2011] Hosom J.P. (2011). Speech Recognition with Hidden Markov Models. Lectures Winter 2011. Oregon Health & Science University, Center for Spoken Language Understanding.
- [Bilmes1998] Bilmes J. (1998) A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. International Computer Science Institute, Berkeley.
- [Young2006] CA. Young, S.J. (2006) HTKBook, 3.4. Cambridge University Engineering Department April, 2006
- [Shinghal1979] Shinghal, R., Toussaint, G.T (1979) Experiments in text recognition with the modified Viterbi algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, 1979
- [Rasmussen2006] Rasmussen, C. E., Williams, C. K. I., Gaussian Processes for Machine Learning, MIT Press, 2006, ISBN 026218253

Chapter 12

Unsupervised Learning



Overview

The concept of unsupervised learning is suitable for finding hidden structures in unlabeled data. They may be informative for the user. Since the examples given to the learner are unlabeled, there is no error penalty or reward signal to evaluate a potential contribution. That means there is no teacher who says that a learning step is correct or incorrect. In some sense HSM techniques are part of unsupervised learning. When dealing with HSM one knows very little about the process. In particular, there is no teacher who confirms the validity of a learning step. Hence unsupervised learning is in the center of interest for HSM. In this chapter we introduce some basic forms of machine learning as far as they are used for our purposes. The focus is mainly on clustering. Clusters are sets containing examples defining a hypothesis that they belong together.

12.1 Generalities

The most common general unsupervised learning methods consist in different types of clustering. Their main purpose is classification. The concept of distances and similarity measurements play a major role. Even if the results are not exact they may provide useful insights. Special systems are self-organizing maps, vector quantization (VQ) and K-means clustering. As properties of the unsupervised learning results:

- The examples are presented in certain structures.
- The user can influence the structure to some degree.
- From the structure of the data one can depict which examples are “close to each other” based on a similarity measure.

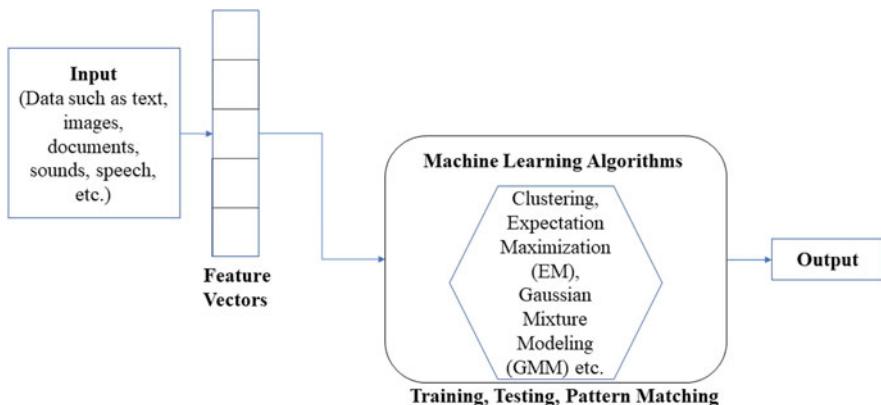


Fig. 12.1 Unsupervised learning

- Examples (signals) are presented without a relation to a class. However, they can be selected in a clever way that say something about the problem area. This can be, for example, presented in certain groups that are related.

There are semi-supervised methods that lay in-between supervised and unsupervised methods. The supervised part can say something, and can provide part of the solution that can be further carried out with unsupervised learning. Typical unsupervised learning is shown in Fig. 12.1.

12.2 Clustering Principles

Definition For a set of objects a clustering is a set of subsets that cover the set. Each subset is called a cluster. For the cluster sets one demands that they cover the given set but they need not be disjoint. Sometimes there is the additional assumption that the clusters are disjoint. Each cluster corresponds to a property that all of its elements have in common. Each cluster set has a center with respect to the defining distance measure. Clusters can be regarded as classes of objects. Often the task of clustering is to discover given classes and their definitions. A success is if the result of clustering coincides with a given set of classes at least approximately. With unsupervised learning, this correspondence will never be known with certainty. The process works iteratively and the use of the following technique is suggested. The principles are:

- Examples are clustered on the basis of some similarity/distance measure. This can be done in different ways. One way is that each example goes to the cluster where it is closest to the center/mean. Another way is that it goes to the cluster where its nearest neighbor is.

- One selects a set of examples as preliminary centers (sometimes called prototypes). They are the initial clusters.
- Each element goes to the clusters where either the center is closest or there is a closest element to it.
- The process of generating clusters is not uniquely prescribed and the notion of “a correct cluster” does not apply.
- However, the classes discovered may be of interest and lead to interesting hypotheses.

Due to the fact that there is no teacher, the classes need to be grouped based on the similarity (or distance) measure. The goal is that in the same cluster objects are close to each other and objects from different clusters are not. The notion of being close is made precise by this similarity/distance measure. However, this procedure implies that different similarity measures can lead to different clusters. Hence, the major task in unsupervised algorithm is defining a useful similarity/distance measure.

12.3 Cluster Analysis Methods

In the clustering problem situation one has the following objects:

- Some set B of training examples.
- Some similarity/distance measure.
- Additional information that has different parameters, e.g. different number k of clusters.

This number k is the input for generating the clusters. However, one does not want to determine k initially. Also, one may have a quality criterion on the clustering. The similarity measure is not directly a quality criterion. This quality criterion measure can be formulated using the measure. Some examples are:

- Prescribe a similarity measure value for the objects in a generated cluster.
- Prescribe the value only for a certain percentage of the cluster elements.

This will lead to a goal of the clustering method. It is the generation of a set of clusters with prescribed properties for instance quality criteria. A general problem is if there are outliers of the clusters. This can occur if the nearest neighbor of some signal is very far away from any center or the other elements of the clusters. The general reason is that the outliers do not belong to any of the intended classes. As a consequence they should not belong to any of the clusters. However, the clusters should cover the space. If there are no more than k clusters allowed, the outliers have to be in one cluster and that may not be intended.

Suppose one has a similarity measure and a quality measure. Then one has a set C_1, \dots, C_k of clusters with the condition that the clustering should be “as good as possible” with respect to a quality criterion of using the “squared errors”. Since computing clustering and its quality measure is deemed crucial we need to build an

intuition for the procedure. There are different kinds of clusters. Possible properties of clusters are:

- Assume real valued examples $b = \{x_1, x_2, \dots, x_n\} \in B$.
- Let C_i be a cluster. The cluster center m_i is the average value of all points of the cluster as defined below in Eq. 12.1

$$m_i = \frac{1}{|C_i|} \sum_{b \in C_i} \quad (12.1)$$

- A quality criterion is the sum of squared distances from the center shown in Eq. 12.2:

$$Q = \sum_{i=1}^k \sum_{b \in C_i} d(m_i, b)^2 \quad (12.2)$$

Here d is some distance measure, e.g. the Euclidean distance. The criterion says that the objects are grouped compactly. A very simple algorithm for cluster generation is provided below:

- (1). Generate all possible clustering for the given examples B .
- (2). Compute for each cluster the quality measure (e.g., Euclidean distance).
- (3). Choose the clustering with highest quality (e.g., lowest global Euclidean distance).

A variation of the method is iterating the procedure if no quality measure is available. There is a serious complexity problem for cluster analysis. The complexity problem provides a major challenge: There are $|B|^k$ possible clusterings! These are far too many to compute. A consequence is that in general optimal clusterings cannot be computed efficiently. Ways out can be for instance:

- (1). Use certain additional assumptions if they are available.
- (2). Find approximate solutions that are easier to compute.

12.4 Special Methods

There are different kinds of methods for clustering. The main methods discussed are *K-Means*, *VQ*, *EM* or *GMM*. Each one requires certain assumptions for working well. They are quite popular and are now explained. Some use additional assumptions while others may give different results.

12.4.1 K-Means

K -means is concerned with the specific unsupervised method of clustering where the number K of clusters is fixed in advance. This occurs for instance if one has a classification problem with a specific number of classes like the number of letters in the alphabet. Each cluster set has a center and it should contain just the objects that are closest to the center. If one starts with arbitrary clusters this will initially not be the case. In K -means one defines a process with an arbitrary initial cluster and improves this step by step. Each step is of the following kind: One computes the **center** of each cluster set. For each object the center is determined which is closest to the object. The object is moved to the set specified by this center. After defining the centers the clusters are recalculated.

K -means minimizes the distortion for a set of vectors o_t for $t = 1, 2, \dots, T$. The objective is to find the set of centers μ_k for $k = 1, 2, \dots, K$ that minimize the distortion. In the K -means, the squared Euclidean distance is mostly used in the clustering process while the expectation-maximization (EM) based GMM uses a mixture resolving approach in the clustering process. The principles can be used for arbitrary distance measures (e.g., Euclidean, Mahalanobis, etc.). In summary K -means has the following steps:

- (1). Set the fixed number of cluster to K
- (2). Initialize the cluster centroids $\mu_1, \mu_2, \dots, \mu_k$ arbitrarily
- (3). Assign the features according to the nearest μ_k
- (4). Recompute $(\mu_1, \mu_2, \dots, \mu_k)$ until there is no significant changes in resulting centroids.

Figure 12.2 shows some steps in the procedure. The figure clearly displays how the elements can move between clusters without changing the number of clusters. In the procedure each step the new centers are computed, The centers are denoted by “+” symbol.

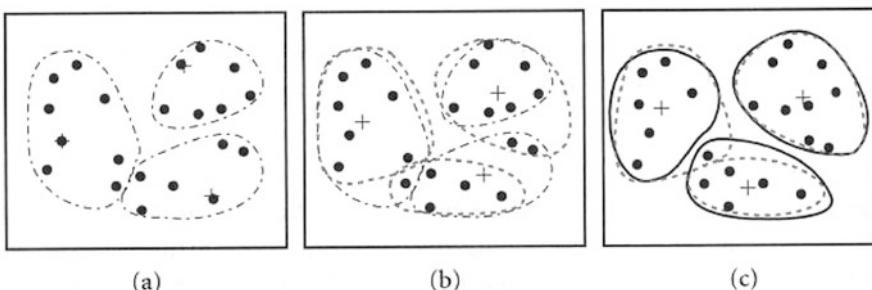


Fig. 12.2 Results of application of K-means procedure

Note here that the centers are not necessarily the given examples. A corresponding requirement would be an extra condition that leads to k -medoid clustering which requires an extra algorithm. k -medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters known a priori. It is more robust to noise and outliers as compared to k -means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances.

- A medoid is defined as the object of a cluster whose average dissimilarity to all the objects of an arbitrary sample point.
- Move the nearest quantization vector center towards this sample point.
- Repeat this until the cluster distortion/distance is minimal. i.e. it is a most centrally located point in the cluster.

12.4.2 Vector Quantization (VQ)

The **Vector Quantization** (VQ) works in unsupervised learning mode using a clustering technique. One assumes that the elements are points in the form of vectors. These are assumptions on the elements and the distance measure. The clusters have weights that used for the classification of images using VQ. The method creates the clusters where the cluster weights are the ratio of points (the vectors) to the cluster of total points in the state. This defines the new clusters at each step. For this one estimates $b_j(o_t)$ by computing means and covariances. $\hat{b}_j(m)$ estimates $b_j(o_t)$ in cluster m . We also use $\hat{b}_j(m)$ for estimating numbers:

$$\hat{b}_j(m) = \frac{\text{Number of vectors in cluster } m \text{ and state } j}{\text{Number of vectors in state } j} \quad (12.3)$$

Then the GMM is applied on the VQ based clusters to model the observations in order to obtain the most likelihood of observations that belongs to a cluster. An input belongs to cluster i if i is the index of the closest center. This has the effect of dividing up the input space into a Voronoi tessellation. A Voronoi tessellation (also called Voronoi regions) is a decomposition of the space in special regions. These regions are determined by a set of centers. This is visualized in given in Fig. 12.3, where the prototype centers are given.

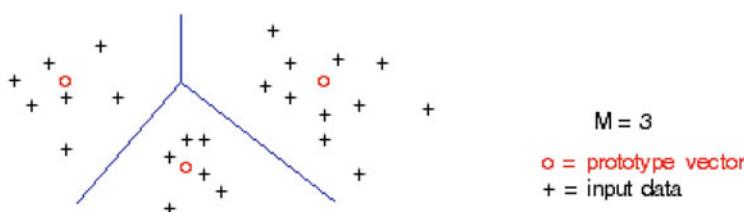


Fig. 12.3 Vector quantization

In Fig. 12.3. we see a tessellation into three regions. Each region is determined by a center prototype vector delineated with red circle.

12.4.3 *Expectation Maximization (EM)*

Suppose you measure a single continuous variable in a large sample of observations. Further suppose that the sample consists of two clusters of observations with different means (and perhaps different standard deviations). Within each sample the distribution of values for the continuous variable follows the normal distribution. We are interested in the resulting distribution of values (in the population). Suppose only the mixture (sum) of the two normal distributions (with different means and standard deviations) would be observed.

We use the **Expectation Maximization (EM)** algorithm given below to obtain the clustering. The goal of *EM* clustering is to estimate the means and standard deviations for each cluster so as to maximize the likelihood of the observed data (distribution). Put in another way, the *EM* algorithm attempts to approximate the observed distributions of values based on mixtures of different distributions in different clusters. The results of *EM* clustering are different from those computed by *k*-means clustering. The latter will assign observations to clusters to maximize the distances between clusters.

The *EM* algorithms do not compute actual assignments of observations to clusters, but rather classification **probabilities**. In other words, each observation belongs to each cluster with a certain probability. This is adequate for stochastic processes of signals. Of course, as a final result you can usually review an actual assignment of observations to clusters, based on the (largest) classification probability.

The EM Algorithm

The term *EM* is short for Expectation Maximization. There are several other algorithms with the same spirit. The expectation is a goal for an optimization problem. The situation is that one has a certain expectation and wants a result that comes close to it. This is a recursive algorithm presented in the usual manner.

Initialization

One starts with some initial probability

$$f_k(x_i | \mu_k, \Sigma_k)$$

denoting the density of object i from the component k with means and variance. A component's contribution is weighted by a mixing coefficient $f(x_i)$ in Eq. 12.4:

$$f(x_i) = \sum_{k=1}^K P_k(f(x_i | k)) \quad (12.4)$$

where $f(x_i | k) = f_k(x_i | \mu_k, \Sigma_k)$.

Induction

The *EM* algorithm is used to estimate the parameters μ_k , Σ_k , and $P(k)$ for the next step. The parameters are somehow initialized (often based on a coarse segmentation using k -means). The *EM*-step updates the parameters by replacing its updated estimate μ_k , Σ_k , and $P(k)$. For N training cases the update equations for the steps are shown in Eq. 12.5:

$$\begin{aligned}\mu'_k &= \frac{\sum_{i=1}^N P'_k(l|x_i)x_i}{\sum_{i=1}^N P'_k(l|x_i)} \\ \Sigma'_k &= \frac{\sum_{i=1}^N P'_k(l|x_i)(x_i - \mu'_k)(x_i - \mu'_k)^T}{\sum_{i=1}^N P'_k(l|x_i)} \\ P'_k &= \frac{1}{N} \sum_{i=1}^N P'(k|x_i)\end{aligned}\tag{12.5}$$

Termination

There is no general automatic stopping criterion. It is provided by the user in each actual situation.

12.4.4 GMM Clustering

We look at the situation where one tries to make use of the fact that one knows something about the objects namely that one faces [Gaussian Mixture Model](#). The method in this situation goes as follows and we have the assumptions and methods:

- (1). There are k components. The i 'th component is called ω_i
- (2). Component ω_i has a mean vector μ_i .
Each component generates data from a Gaussian with mean μ_i and covariance matrix σ_i
- (3). Assume that each data point is generated according to the following method by picking a component at random:
 - (a). Choose component i with probability $P(\omega_i)$.
 - (b). Choose a data point $N(\mu_i, \omega_i)$.

There are k components. The i 'th component is called ω_i .
- (4). Component ω_i has a mean vector μ_i

Each component generates data from a Gaussian with mean μ_i and covariance matrix σ_i . This makes use of the Gaussians. Assume that each data point is generated according to the indicated method that utilizes the GMM property. It gives a cluster as before.

12.5 Background Information

General

Unsupervised learning tries to find hidden structures in unlabeled data, see Hastie et al. (2009). This may result from the fact that the corresponding knowledge for principle reasons does not exist or that one has presently no access to it. The main element in unsupervised learning is that during the learning phase there is no teacher available. This points for using it in the direction of HSM. Clustering is a very popular unsupervised method. It has several variations.

Past

The term “k-means” was first used by James MacQueen in 1967, (MacQueen 1967). K-means can be found in Theodoridis and Koutroumbas (2009) and MacKay (2003). Vector quantization (VQ) is a quantization technique. It is a classical quantization technique used for data compression which allows the modeling of probability density functions by the distribution of prototype vectors. It was originally used for data compression. Today it is used for lossy data correction, pattern recognition, density estimation and clustering.

More on vector quantization is in Gray (1984). See also Arthur (2007), Theodoridis and Koutroumbas (2009).

Suggested

For the EM algorithm we recommend Gupta, M.R., and Chen, Y. (2010). It uses optimization techniques. It contains also the use of it to HSM. Even if the probabilities are hidden one may still have expectations. In fuzzy clustering the same principles as in the general fuzzy approach are used. The data elements can belong to more than one cluster, and associated with each element is a set of membership levels. It can be found in Bezdek (1981). Processes with intervals as values are natural candidates for fuzzy treatments. Such processes with clustering techniques are discussed there too.

Readers are suggested to read references used in this chapter such as Varghese and Wilscy (2013).

12.6 Exercises

Exercise 1 Give an example of data points in the real plane such that clusters obtained with K-Means lead to an unwanted result if outliers are present.

Exercise 2 Give an example with outliers where VQ is not adequate.

Exercise 3 Compare for fuzzy sets the fuzzy clustering with the method that first applies defuzzification and then clustering.

Exercise 4 Implement a method for clustering where K-means is insufficient. What would you recommend instead?

References

- [MacQueen1976] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press.
- [Hastie2009] Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning Data Mining, Inference, and Prediction, New York, Springer.
- [Theodoridis2009] Theodoridis and K. Koutroumbas (2009). Pattern Recognition. Academic Press, Elsevier, UK, 4th edition.
- [MacKay2003] MacKay, D (2003). Information Theory, Inference and Learning Algorithms. Cambridge University Press.
- [Gray1984] Gray, R.M. (1984). “Vector Quantization”. IEEE ASSP Magazine 1 (2).
- [Varghese2013] Varghese, E., Wilscy, M. (2013). Face Recognition Based On Vector Quantization Using Fuzzy Neuro Clustering International Journal of Computer, Control, Quantum and Information Engineering Vol:7
- [Arthur2007] Arthur, D., Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027–1035). Society for Industrial and Applied Mathematics.Theory and Use of the EM Algorithm. Foundations and TrendsR in Signal Processing, Vol. 4, No. 3.
- [Bezdek1981] Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. ISBN 0-306-40671-3.
- [Gupta2010] Gupta, M.R., Chen, Y. (2010). Theory and Use of the EM Algorithm, Foundations and Trends in Signal Processing, Vol. 4, No. 3, 2010

Chapter 13

Markov Model and Hidden Stochastic Model



Overview

A general categorisation in the realm of HSM is that one distinguishes between stationary and non-stationary processes. The stationary processes have the same statistical characteristics regardless of the time shifts along the axis. That means the parameters of the probability model of the process are time-invariant. Markov models have been investigated in Chap. 10, and here we will extend that discussion. Markov models are specific stochastic models. Hidden Markov Models deal with an even more specific form of stochastic process modelling where the states are not directly observable. In this chapter we first introduce the basic definitions of Markov models and HSM and then we provide examples. The problem description has three stages:

- (1). There is an evaluation which says the certainty of the model fits to the data. That means, the model describes the data or features well.
- (2). Decoding this task involves estimating the hidden states based on the observed data or features, and Learning. This gives the probability of how the model fits to the data and the probability of the hidden states based on the observed variables.
- (3). An extension is explored for the Gaussian Mixture Models, GMM.

13.1 Markov Process

Often, the probability distribution of the events is at most approximately known. Such situations are considered in the area of Hidden Stochastic Models (HSM). A point is that we here consider arbitrary stochastic processes and not only Markov ones. The main task for HSM is to determine the statistical parameters by looking at examples. That means one applies machine learning. Next we will show how some major signal processing methods are applied to machine learning.

13.2 Gaussian Mixture Model (GMM)

The [Gaussian Mixture Model](#) extends considerations discussed in the previous chapter. It means that one has G many Gaussian functions involved, where G is a natural number. The *GMM* is a mixture of a number Gaussian models. It is frequently used for HSM as it is a quite general approximation of probability density functions. This mixture can be overlapping; meaning that the data points can be associated, with different probabilities, to different competing explanations. It is commonly used to model an arbitrary signal space. The GMM is, in the first place, a representation method taking care of such aspects. It is a linear combination of $G \in \mathbb{Z}$ many Gaussians. However, there are special methods connected to it.

This Fig. 13.1 visualizes a Gaussian Mixture (depicted in red in the figure) and one sees the different Gaussians (depicted in blue).

Given the vectors, each Gaussian model in the GMM has its own mean and covariance matrix as its parameters and these have to be estimated separately for each Gaussian model for the GMM. The computation of the mean and covariance matrix for Gaussian components of a GMM is not performed in the same manner that is done for a stand-alone Gaussian model because here we do not know which observation belongs to which Gaussian, i.e., which mean and covariance matrix belongs to which feature vectors.

The GMM combines probability distributions by some weighting. These weights are unknown for a particular application and these weights have to be determined. These determinations are done by an estimation. This estimation is performed via an iteration process for a GMM in order to show the recursive nature of the model.

Initialization

The initial weights are taken in such a way that they are all the same. This means for the G Gaussian models one has to determine G many weights. The weights are initialized to $\frac{1}{G}$. The GMM model parameters are initialized by K-means clustering as discussed in earlier chapters.

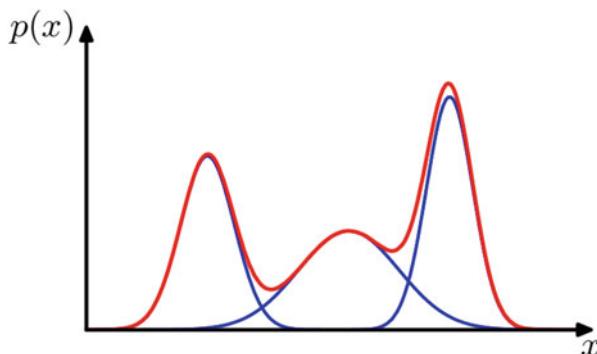


Fig. 13.1 Gaussian Mixture example

Induction

The goal is to optimize the final model parameters. This is done by applying the expectation maximization (EM) algorithm. The EM is an iterative process using s in Eq. 13.1 as depicted below where i denotes the number of the iteration. The GMM provides a smooth, overall distribution fit. Its components detail the multi view.

$$Q(\lambda^i, \lambda^{i+1}) = \sum_{t=1}^T \sum_{q=q_1}^{q_N} p(q|o_t, \lambda^i) \log p(q|o_t, \lambda^{i+1}) \quad (13.1)$$

The EM iteration process was discussed above.

Termination

The termination is the same as for EM.

13.3 Advantages of Using GMM

The GMM acts as a hybrid model between unimodal Gaussian and the vector quantization clustering process by using a discrete set of Gaussian functions, each with their own mean and covariance matrix to allow a better modeling capability. One of the powerful properties of the GMM is its ability to form smooth approximations to arbitrarily shaped densities.

13.4 Linear Prediction Analysis

The goal of Linear Prediction Analysis is to derive quality models for each class of sound source. It can be shown that solution equations for both deterministic and stochastic signal classes are similar in structure. This solution approach is referred to as linear prediction analysis and works for both:

- Deterministic: Speech Sounds with periodic or impulse sources
- Stochastic: Speech Sounds with noise sources

Linear prediction analysis leads to a method of speech analysis based on the all-pole model in Eq. 13.2:

$$H(z) = \frac{A}{1 - \sum_{k=1}^P a_k z^{-k}} \quad (13.2)$$

The intention is to estimate filter coefficients $\{a_1, a_2, \dots, a_P\}$; for a particular order P , and gain A , over a short-time span of speech signal (typically 20 ms) for which the signal is considered quasi-stationary. One can use the linear prediction method

where each speech sample is approximated as a linear combination of past speech samples. This in turn lead to the set of analysis techniques for estimating parameters of the all-pole model as shown in Eq. 13.3.

$$H(z) = \frac{A}{1 - \sum_{k=1}^P a_k z^{-k}} = \frac{S(z)}{U_g(z)} \quad (13.3)$$

In time domain this expressions can be rewritten by Eq. 13.4:

$$s(n) = \sum_{k=1}^P a_k s(n-k) + A u_g(n) \quad (13.4)$$

where with A we denote the gain of the signal, P is the order of the model, $u_g(n)$ is the driving input signal, the a_k parameter vector is to be estimated, $s(n-k)$ are known past samples, and finally, $s(n)$ is the next sample to be predicted. The method used to predict a current sample from a linear combination of past samples is called linear prediction analysis.

The quantization of linear prediction coefficients or of a transformed version of these coefficients is called linear prediction coding (LPC). If we assume that our driving signal is $u_g(n) = 0$ then the original sequence can be expressed by Eq. 13.5:

$$s(n) = \sum_{k=1}^P a_k s(n-k) \quad (13.5)$$

The prediction error sequence is given as the difference between the original sequence $s(n)$ and its prediction $\hat{s}(n)$:

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^P \alpha_k s(n-k) \quad (13.6)$$

where with α_k we have indicated the (unknown) model parameter values. The goal of this approach is to have those model parameters $\{\alpha_k\}$ be identical to actual model parameters $\{a_k\}$.

An important question is: how to derive an estimate of the prediction coefficients a_k , for a particular order P , that would be optimal in some sense. Optimality is measured based on a criteria. An appropriate measure of optimality is the mean-squared error (MSE). The goal is to minimize the mean-squared prediction error: E is formulated by Eq. 13.7:

$$E = \sum_{m=-\infty}^{+\infty} e^2(m) = \sum_{m=-\infty}^{+\infty} [s(m) - \hat{s}(m)]^2 \quad (13.7)$$

In reality, a model must be valid over some short-time interval, say M samples on either side of n as depicted in Fig. 13.2.

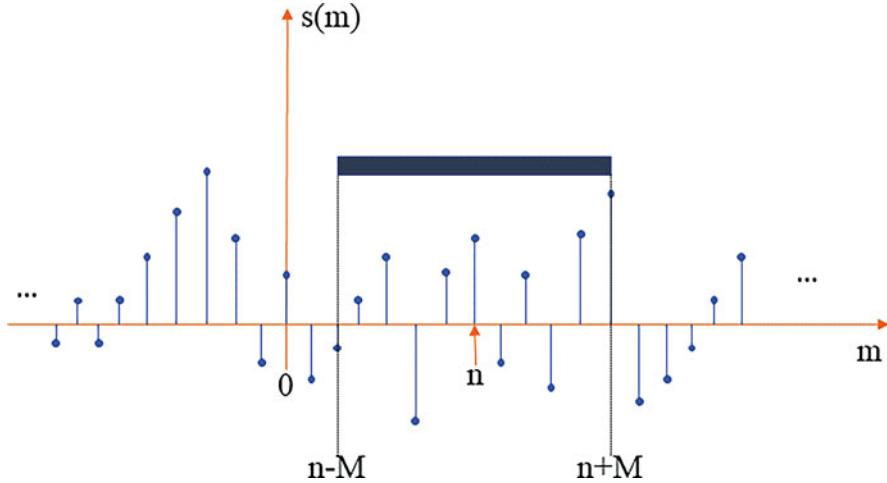


Fig. 13.2 Short-time prediction widow

The method applied for determining unknown parameters is the mean-squared error as applied here. The goal is to determine a set of unknown parameters $\{\alpha_k\}$ for which E is minimal. Assuming quadratic surface, the parameters can be found from the set of Eqs. 13.8.

$$\frac{\partial E}{\partial \alpha_i} = 0 \quad i = 1, 2, \dots, P \quad (13.8)$$

This leads to Eq. 13.9

$$\begin{aligned}
 \frac{\partial E}{\partial \alpha_i} &= \frac{\partial}{\partial \alpha_i} \sum_{m=-\infty}^{+\infty} \left(s(m) - \hat{s}(m) \right)^2 \\
 &= \frac{\partial}{\partial \alpha_i} \sum_{m=-\infty}^{+\infty} \left(s(m) - \sum_{k=1}^P \alpha_k s(m-k) \right)^2 \\
 &= \sum_{m=-\infty}^{+\infty} \frac{\partial}{\partial \alpha_i} \left(s(m) - \sum_{k=1}^P \alpha_k s(m-k) \right)^2 \\
 &= \sum_{m=-\infty}^{+\infty} 2 \left(s(m) - \sum_{k=1}^P \alpha_k s(m-k) \right) \left(-\frac{\partial}{\partial \alpha_i} \sum_{k=1}^P \alpha_k s(m-k) \right) \\
 &= \sum_{m=-\infty}^{+\infty} 2 \left(s(m) - \sum_{k=1}^P \alpha_k s(m-k) \right) \left(-s(m-i) \right) \\
 &= \sum_{m=-\infty}^{+\infty} 2 \left(-s(m)s(m-i) + \sum_{k=1}^P \alpha_k s(m-k)s(m-i) \right) = 0
 \end{aligned} \quad (13.9)$$

Equation 13.9 can be rewritten by multiplying through Eq. 13.10:

$$\sum_{m=-\infty}^{+\infty} s(m)s(m-i) = \sum_{k=1}^P \alpha_k \sum_{m=-\infty}^{+\infty} s(m-i)s(m-k) \quad i = 1, 2, \dots, P \quad (13.10)$$

This last equation is referred to as the normal equations. Several remarks on that derivation are in order:

- (1). Order (P) of the actual underlying all-pole transfer function is not known.
 - Order can be estimated by observing the fact that a P th order predictor in theory equals that of a $(P + 1)$ order predictor.
 - Also predictor coefficients for some $k > P$ equal zero (or in practice close to zero and model only models random-noise effects).
- (2). Prediction error $e(m)$ is non-zero only “in the vicinity” of the time n : $[n - M, n + M]$.
 - In predicting values of the short-time sequence $s(m)$, P -values outside the prediction error interval $[n - M, n + M]$ are required. This fact gives raise to two methods.
 - **Autocorrelation** method - This method assumes that speech samples are zero outside the interval and hence those samples are not used nor needed.
 - **Covariance** method - In this method the values outside the interval predict values inside the interval.

We can formulate the general Linear Prediction in an expanded matrix formulation using Eq. 13.11 as follows:

$$\begin{bmatrix} s(n - M + 0 - 1) & s(n - M + 0 - 2) & \dots & s(n - M + 0 - P) \\ s(n - M + 1 - 1) & s(n - M + 1 - 2) & \dots & s(n - M + 1 - P) \\ \vdots & \vdots & \ddots & \vdots \\ s(n + M - 1) & s(n + M - 2) & \dots & s(n + M - P) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_P \end{bmatrix} = \begin{bmatrix} s(n - M + 0) \\ s(n - M + 1) \\ \vdots \\ s(n + M + 0) \end{bmatrix} \quad (13.11)$$

The approach presented is very general and does not dive into efficiency and conditions under which the samples are collected. The methods developed are recognized depending if the samples outside the prediction interval are used for prediction or not. If the samples are used outside the prediction interval, that approach gives rise to so called **Covariance** method. That approach will give a correct answer if the matrix presented previously is invertable. Unfortunately, this turns out to be the case because the methods is very sensitive. That is particularly the case if the source of the sound produced is generated nonlinearly (e.g shouting, etc.). Note that the presented method assumes linearity and if this is not the case the general (**Covariance** method) approach will fail.

The alternative is not to use the samples outside the prediction widow. This in turn leads to the approach that is referred as **Autocorrelation** methods. However, this methods is suboptimal and only can give an approximate solution. On the other hand it does not have problem with stability of inversion of the matrix and it leads to a stable solution for which there is a efficient algorithm presented next.

13.4.1 Autocorrelation Method

The autocorrelation is a very common and general approach. It is essentially close to the Yule-Walker approach, see the details presented below. This method is one of the most commonly used for the model parameter estimation. The autocorrelation approach is simple and efficient. This method is suboptimal, however it leads to an efficient and stable estimation procedure. Assumes that the samples outside the prediction time interval $[n - M, n + M]$ are all zero, and extends the prediction error interval, i.e., the range over which we minimize the mean-squared error to $\pm\infty$.

Short-Time Interval $[n, n + N_w - 1]$ where $N_w = 2M + 1$ (Note: it is not centered around sample n). Segment is shifted to the left by n samples so that the first nonzero sample falls at $m = 0$. This operation is equivalent to: Shifting of speech sequence $s[m]$ by n -samples to the left and windowing by N_w -point rectangular window:

In the autocorrelation approach, the number N of non-zero samples of a certain length l is nonzero, and it is zero outside of the length l . The approach is the cross-correlate the signal with itself. The autocorrelation of a stochastic process describes the correlation between values of the process at different times, as a function of the time lag. It is a method for finding repeating patterns. Examples are the presence of a signal disturbed by noise, or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. If time is involved, then the average d of the autocorrelation function is replaced by the time averaged autocorrelation function. Given a signal $f(t)$ the continuous autocorrelation $R_{ff}(\tau)$ is most often defined by Eq. 13.12 as the continuous cross-correlation integral of $f(t)$ with itself, at lag τ :

$$R_{ff} = f * g_{-1}(f)(\tau) = \int_{-\infty}^{+\infty} f(u + \tau) f(u) du \quad (13.12)$$

where f is the complex conjugate of f , and g_{-1} is a function which manipulates the function f and $g_{-1}(u) = f(-u)$ represents convolution. The parameter u in the integral is a dummy variable and is only necessary to calculate the integral. It has no specific meaning. A basic property of the autocorrelation is symmetry in Eq. 13.13 presented below:

$$R_f(i) = R_f(-i) \quad (13.13)$$

The definition of the autocorrelation between times s and t is provided in Eq. 13.14:

$$R_f(s, t) = \frac{E[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s} \quad (13.14)$$

To describe this further, we now suppose f is the signal function that is one-dimensional. Then a property of the autocorrelation is symmetry as presented in Eq. 13.15, the autocorrelation is an even function:

$$R_f(-\tau) = R_f(\tau) \quad (13.15)$$

Now we give a formulation of the linear prediction (LP) for the all pole modeling. The LP approach uses a least squares approximation that gives a set of linear equations in order to find an approximate solution for the auto-regressive (AR) parameters. The LP approach is derived using an auto-correlation approach. This is close to the Yule-Walker approach. In the LP analysis the current sample is modeled by linear combinations of its P most recent past samples. P is the prediction model order. This realized the earlier remark that signals are dependent on earlier signals. The LP analysis uses mean squared error criteria for the best predicted samples such that it is closest to the real sample.

There are some assumptions of using LP. First we mention the assumptions on signal modeling. They are used in the LP based speech (several methods are for speech only):

- (1). The excitation source and the vocal-tract system are independent from each other.
- (2). Each excitation actuates at the beginning of each segment and remains active until the end of the segment.
- (3). The vocal tract system is modeled using the AR process where the excitation source is white noise. The vocal-tract changes its shape slowly. Its characteristics changes at every 10 to 30 ms time interval.

The parameters are computed at each short time interval at each 3 to 10 ms intervals. The order of the model is equal to the number of the parameters that represent the speech. We explain in part C the relation between the model order and vocal tract tube and how the model order and parameters are related. The order has to be large enough to represent the coefficients needed.

13.4.2 Yule-Walker Approach

Instead of using a correlation approach, the Yule-Walker approach to obtain the auto-correlation matrix is applied. This method is a most straight forward one for the AR model parameters. In this approach, the ensemble autocorrelation $r[i]$ is replaced by the corresponding time-averaged autocorrelation computed from a

given block of data. The minimization of the mean squared error and the use of a windowed signal is central. Before, we have explained how we behave when we have p many linear equations. The LP solution using this approach needs the inversion of the transition matrix and the multiplication of a $p \times p$ matrix with a $p \times 1$ length r vector. A certain form of a matrix that is used is the Toeplitz matrix. It allows a simplified approach to the problem.

Definition A Toeplitz matrix is a matrix in which each descending diagonal from left to right is constant. That means a matrix \mathbf{A} can be expressed by Eq. 13.16.

$$\mathbf{A} = (a_{ij}) \quad (13.16)$$

is called a Toeplitz-matrix if its entries a_{ij} given in equation above depend only from $i - j$ of the indices. For linear equations with Toeplitz matrices there are special efficient solution algorithms.

The matrix is a symmetric matrix and thus we have $r(i, k) = r(k, i)$. The solution for the parameters can be obtained by using a Gaussian elimination approach or by the Levinson-Durbin recursion. Levinson-Durbin recursion is a procedure in linear algebra to recursively calculate the solution to an equation involving a Toeplitz matrix. The Levinson approach is efficient in solving for the parameters of a Toeplitz matrix.

13.4.3 Covariance Method

In the covariance approach, the minimum mean squared error is computed using the derivative of equation presented below with respect to a_k for $k = 1, 2, \dots, P$. The starting point is the mean squared error formulated by Eq. 13.17.

$$E = \frac{1}{N} \sum_{n=0}^{N-1} \left(s(n) - \sum_{i=1}^P \alpha_i s(n-i) \right)^2 \quad (13.17)$$

The summation limit can be any point starting from $n = p$ to $n = N$. Here the truncation of the signal is not essential and an explicit signal windowing is not done. Therefore, in this approach the spectral distortions from the rectangular windowed signal do not occur. Instead of when using a correlation approach applying the Yule-Walker is to obtain the autocorrelation matrix. Here in the covariance approach, the covariance matrix C is derived from, if the equation is positive definite and symmetric but it is not Toeplitz. In Eq. 13.18 one has $n = 0, 1, \dots, N-1$ and $i, j = 1, 2, \dots, p$.

$$c_n(i, 0) = \sum_{j=1}^P \alpha_j c_n(i, j) \quad (13.18)$$

An extended version of the Eq. 13.18 for $c_n[i; k]$ is shown in Eq. 13.19.

$$c_n(i, k) = \sum_{j=l}^{l+N-1} s(n-i)s(n-k) \quad (13.19)$$

This equation reflects P linear equations that can be written in a matrix vector form of Eq. 13.20:

$$\begin{bmatrix} c(1, 1) & c(1, 2) & \cdots & c(1, P) \\ c(2, 1) & c(2, 2) & \cdots & c(2, P) \\ \vdots & \vdots & \ddots & \vdots \\ c(P, 1) & c(P, 2) & \cdots & c(P, P) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_P \end{bmatrix} = \begin{bmatrix} c(1, 0) \\ c(2, 0) \\ \vdots \\ c(P, 0) \end{bmatrix} \quad (13.20)$$

Presented equation can be written in a compact form with Eq. 13.21

$$\mathbf{c} = \mathbf{C}\boldsymbol{\alpha} \quad (13.21)$$

Finding the solution to the LP parameters $\boldsymbol{\alpha}$ requires computing the inverse matrix \mathbf{C}^{-1} . The covariance matrix \mathbf{C} is symmetric, positive definite but not Toeplitz, therefore Levinson Durbin recursion is not used, instead the Cholesky decomposition is used for the parametric solution. In Fig. 13.3 we see the analysis of the signal model using the covariance approach. We number the parts clockwise alphabetically, starting with a in the upper left. In a , we see the segment of the signal. In c , we see the frequency response of the LP filter with parameters extracted by the covariance approach. In b , we see the pole position of the covariance approach and in d , we see the excitation which is the output of the filter. This is white noise.

13.4.4 Comparison of Correlation and Covariance Methods

These solutions to the linear prediction normal equations are using these methods that have matrix with defined different properties. It depends on the situation whether they are strong or weak. The dependency is mainly related to properties of the signal under consideration. When signals are long two different solutions are virtually identical. In the autocorrelation method this is somewhat easier to compute because of the greater redundancies. Experimental evidence indicates that the covariance method is more accurate for periodic sounds, while the autocorrelation method performs better for fricative (noise like) sounds.

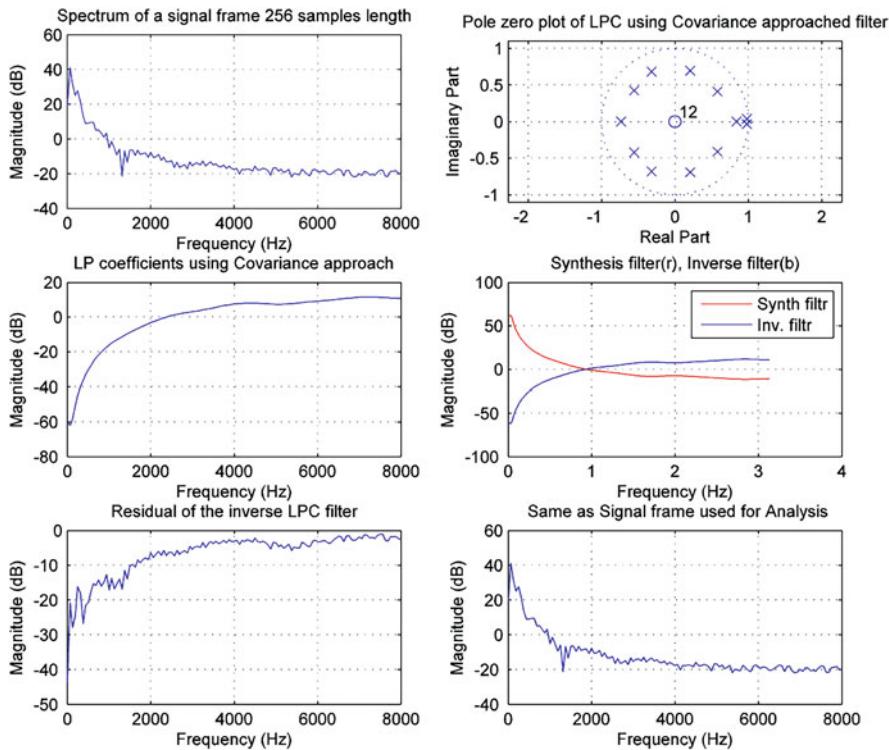


Fig. 13.3 Covariance method applied to a frame of speech

13.5 The ULS Approach

In this approach the error is minimized by computing the average of the sum of the squares of the estimated forward and backward linear reduction errors. The forward and backward prediction errors are computed in order to compute their combined error. In Eq. 11.24, we have seen the backward prediction coefficients are obtained as the reverse version of the forward prediction coefficients.

The **Unweighted Least Square** (ULS) approach is a modified covariance method. This is based on respect to all the prediction coefficients, whereas the Burg method performs a constrained least squares minimization with respect to only a single prediction coefficient J introduced in Eq. 13.2. This represents $p + 1$ by $p + 1$ dimensional reflection matrix and 0 denotes transpose. Using the reflection matrix, we get a relation between forward linear prediction and backward linear prediction. This is shown in Eqs. 13.23 and 13.24 presented below.

$$J \approx [\beta_1^p, \dots, \beta_p^p] = [\alpha_p^p, \dots, \alpha_1^p] \quad (13.22)$$

We get the forward prediction error $efp[n]$ and backward prediction error $ebp[n]$ in Eqs. 13.23 and 13.24. The total $N - p$ forward linear prediction error elements and the $N - p$ backward linear prediction error elements can be formed from N data samples without searching through all the available data. They have been introduced above; a slightly other description is:

$$\text{FLP error : } e_p^f[n] = s[n] - \hat{s}[n] = s'_p[n] \alpha_p^{fb} \quad (13.23)$$

$$\text{BLP error : } e_p^b[n] = s[n - p] - \hat{s}[n - p] = s'_p[n] J \alpha_p^{fb} \quad (13.24)$$

Figure 13.3 explains how the signal is processed for the analysis. Initially the analog speech waveform $s(t)$ is digitized into $s[k]$ where $k = 0, 1, 2, 3, \dots, K - 1$. These are blocked into segments s_m with $m = 1, 2, \dots, M$. Each block s_m has N many samples indicated by $n = 0, 1, 2, \dots, N - 1$ and each current sample $s[n]$ is modeled by linear combinations of its previous p many samples. p is the prediction or model order. This is written in Eq. 13.25 depicted below:

$$\begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ s[K - 1] \end{bmatrix} \xrightarrow{\text{Segmented Signals}} \begin{bmatrix} s_1[0] & s_1[1] & \cdots & s_1[N - 1] \\ s_2[0] & s_2[1] & \cdots & s_2[N - 1] \\ \vdots & \vdots & \ddots & \vdots \\ s_M[0] & s_M[1] & \cdots & s_M[N - 1] \end{bmatrix} \quad (13.25)$$

Thus in Eq. 13.25 s is $M \times N$ dimensional. Generally a signal is blocked using a window function. In such case the signal is multiplied by a window function. Figure 13.4 shows the various effects of ULS as in example of Pole-Zero plot of ULS based LP filter (Figs. 13.5 and 13.6).

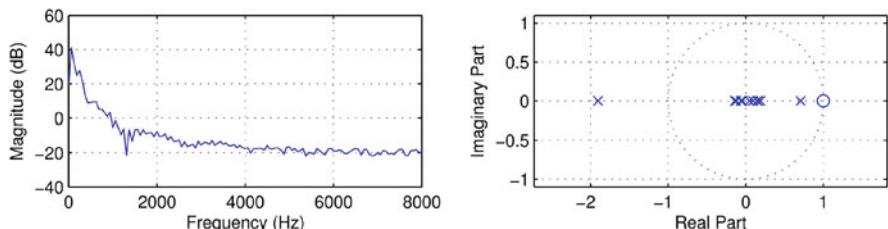


Fig. 13.4 ULS based Prediction Coefficients and pole-zero locations of the filter and inverse filter

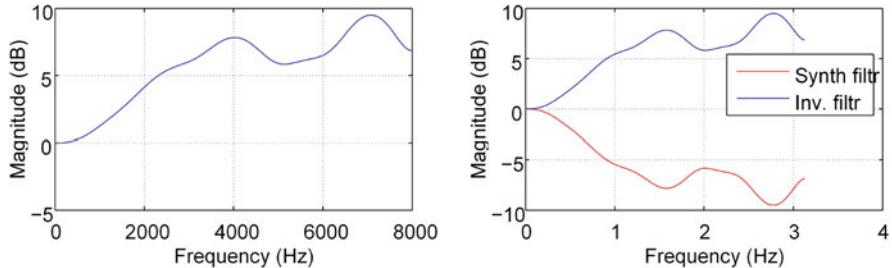


Fig. 13.5 Residual of the inverse ULS filter

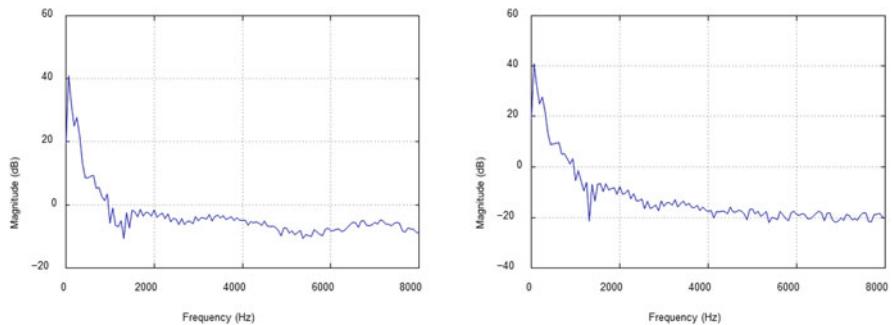


Fig. 13.6 Frequency response of ULS and covariance approach

13.6 Comparison of ULS and Covariance Methods

The windowing is generally used to control the effect of the edges by providing tapering of side lobes in the spectral estimation. A typical length of the window function is equal to the length of the signal block. However, in a covariance, Burg or ULS based LP approach, a windowing of a signal is not necessarily needed. Both, the Burg approach and the modified covariance algorithm are based on the minimization of the forward and backward squared prediction errors (see below sections where this topic is discussed). The ULS approach is based on the minimization of the prediction coefficients. The Burg approach sets constraints on the LP coefficients so that these coefficients satisfy the Levinson recursion conditions and obtain least squares optimization using reflection coefficients in order to solve AR parameters problems. Some problems such as spectral line splitting, bias of the frequency estimates are eliminated in ULS approach. The only problem applying the ULS approach is its weakened stability issue of the LP coefficients but mainly it does not appear when the ULS is structured following stable lattice filters which is used here. Next we summarize some important aspects and criticisms.

- The ULS approach and Burg approach can be analyzed using the lattice structure. This is useful to capture the physical speech production process efficiently.
- The Yule-Walker introduces poor estimated parameters.
- The problems such as line splitting, frequency bias, and spurious or false peaks are observed in the Burg approach.
- The ULS approach may result in instability where the Yule-Walker approach and the Burg approach may generate stable model analysis. In spectral estimation a model stability is not a major concern.

13.7 Forward Prediction

In a given model the probability of the set of observations in a specific state sequence has to be estimated. Given the model, the probability of being at state i at time t can be estimated by the forward probability denoted by α . The forward probability can in principle be computed with the recursion by using Eq. 13.26 provided below:

$$\alpha_t(i) = p(\mathbf{o}, \mathbf{q} | \lambda) = p(o_1, o_2, \dots, o_i, q_t = i, \lambda) \quad (13.26)$$

The forward algorithm estimates the likelihood $p(\mathbf{o} | \lambda)$ (which means for the given model the probability of the observation at a certain time and at certain state) by the following recursion:

Initialization

The procedure uses the initial parameters to start the evaluation as is shown in Eq. 13.27 for $t = 2, 3, \dots, T$ and $j = 2, 3, \dots, N$. This gives the estimate of the likelihood:

$$\alpha_1(i) = \pi_i b_i(o_1) \quad \text{for } 1 \leq i \leq N \quad (13.27)$$

Induction

The probabilities of the observations from the past to the present state are computed using the previous probabilities, transition probabilities, and observation probabilities in order to estimate the likelihood of the observations. The computation of the observation for a state is the modeled by Eq. 13.28 presented below, where T is the length of the each feature observation:

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t) \quad \text{for } 1 \leq t \leq T \text{ and } 1 \leq j \leq N \quad (13.28)$$

Termination

This gives the estimate of the likelihood of the observation for a state given the model by Eq. 13.29 where T is the length of the each observed feature.

$$\alpha_T(n) = p(\mathbf{o}|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (13.29)$$

13.8 Backward Prediction

The backward search is done by the so called backward algorithm that indicates direction of search. The backward probability β denotes the probability of the observations o_t through o_{t+1} being in state i at time t given a HSM model λ utilizing the Eq. 13.30. Here the probability of the future sequence conditioned on the present state j at time t is computed.

$$\beta_t(i) = p(o, q|\lambda) = p(o_{t+1}, \dots, o_T, q_t = i|\lambda) \quad (13.30)$$

Initialization

The model in state i at time T is 1. The transition of the observation is finished at time $T + 1$ and this is also formulated in Eq. 13.31.

$$\beta_T(i) = 1 \quad \text{for } 1 \leq i \leq N \quad (13.31)$$

Induction

This step gives the likelihood of the observation given the model. It is shown in Eq. 13.32.

$$\beta_t(i) = \sum_{j=1}^N b_j(o_{t+1}) a_{ij} \beta_{t+1}(j) \quad \text{for } 1 \leq t \leq T \text{ and } 1 \leq i \leq N \quad (13.32)$$

Termination

This provides with the procedure how to end the backward procedure. Suppose β_0 is the state at the beginning signal that emits the π values in the transition to the first states at time 1. Then we get Eq. 13.33.

$$\beta_0(1) = \sum_{j=1}^N \pi_j b_j(o_1) \beta_1(j) \quad (13.33)$$

13.9 Forward-Backward Prediction

This approach combines the prediction kinds just introduced. The point is here that one does not get essentially new predictions. The expectation is rather to obtain a higher efficiency in the prediction process. For the combination one needs many examples. A particular (and recommended) method is the Baum-Welch algorithm.

13.10 Baum-Welch Algorithm

This is used for training the model by the Baum-Welch technique which uses the forward-backward algorithm and using the model parameters that are obtained from the observation sequences. The general idea is expressed in an obvious quotient (e.g., transition probability of moving from state s_j to state s_i) depicted in Eq. 13.34:

$$P(s_i|s_j) = \frac{\text{Expected number of transitions from state } s_j \text{ to state } s_i}{\text{Expected number of transitions from state } s_j} \quad (13.34)$$

The combination of the forward and backward probability is computed by relation provided below:

$$p(o, q_k = i | \lambda) = \alpha_t(i) \beta_t(i) \quad (13.35)$$

This is used to find the model parameters from the forward and backward direction in the frame of EM algorithm. It stops when the likelihood is the same for both the forward and backward algorithms. The Baum-Welch technique uses the following steps for training the model and the model parameters of the observation sequences:

- Compute the forward probabilities α by using the forward algorithm.
- Compute the backward probabilities β by using the backward algorithm.
- Compute the transition probabilities A and the emission probabilities B at the current state using the observation sequences.
- Compute the new model parameters μ , Σ and c .
- Compute the new log likelihood of the model.
- Stop computations when there is no change in the log-likelihood.

The method is applied to the *GMM* model. For the *GMM* model parameters estimations, the solutions for the re-estimation formula for \hat{c} , \hat{u} , and $\hat{\Sigma}$ are estimated for the observations $b_j(o_t)$. Baum-Welch estimation technique is outlined below:

Initialization

It starts with the probability of the model given at a certain state i at time t . It is shown by Eq. 13.36 next:

$$p(o, q_k = i | \lambda) = \alpha_t(i) \beta_t(i) \quad (13.36)$$

Induction

Apply the forward and the backward methods to Eq. 13.36.

Termination

It stops learning when the likelihood is the same for both the forward and backward algorithms. The Baum-Welch algorithm works for any stochastic process. Technically it is similar to the Viterbi algorithm. Originally it was developed for speech recognition but it is of more general character and so it is presented here.

13.11 Viterbi Algorithm

In the Viterbi search, the best state sequence along a single path at a certain time t and the best scores are computed. This process is simplification to Baum-Welch procure where all scores are computed and kept. This is in contrast to Viterbi where only the best scores are computed at each time instance as stated earlier. The highest likelihood of observation sequence of input feature vector $o_t(i)$ of the state i at time t it is given by equation presented below, Eq. 13.37. Notation used to track the wining sequences of states is $\Psi_t(j)$ that defines the best state prior to stat j at time t . The a_{ij} denotes the transition probability from state i to state j in the transition matrix A .

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} p(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, o_3, \dots, o_t | \lambda) \quad (13.37)$$

and

$$\Psi_t(i) = \operatorname{argmax}_{q_1, q_2, \dots, q_{t-1}} p(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, o_3, \dots, o_t | \lambda) \quad (13.38)$$

13.12 Background Information

General

GMM is short for Gaussian Mixture Model. It corresponds to the mixture representation of several Gaussian probability distributions of observations in the overall population. See McLaughlan (2000). The correlation describes in general the relation of two variables to each other. For the autocorrelation one considers the relation to itself. Covariance is a measure of how much two random variables change together. The sign of the covariance therefore shows the tendency in the linear relationship between the variables.

Past

ULS is short for Unified Logging Service. The covariance method and the ULS approach are very well discussed in Marple (1987a) and Marple (1987b). For correlation see Rodgers and Nicewander (1988). Remarks on the Yule-Walker approach are in Hazas (1999). Spectral information k is mainly used for images. For the spectral estimation see Marple (1987a).

Suggestion

Autocorrelation is the cross-correlation of a signal with is the similarity between observations as a function of the time lag between them. It is used for finding repeating patterns, such as the presence of a periodic signal obscured by noise. See for instance Box et al. (1994). Computational advices are given in www.mathworks.com/help/econ/autocorr.html and www.mathworks.com/.../test-for-autocorrelation.html. Another source is Dunn (2005). An EM (Expectation Maximization) algorithm is a method for computing the most likely estimates of parameters in statistical models in HSM. In HSM there is no given probability what makes this very difficult. One of the early papers on EM was Dempster et al. (1977). Details on the EM algorithm are in Bilmes (1998). The Toeplitz matrices have several computational advantages in the present context and therefore they are used as often as possible. An early reference on Toeplitz matrices are in Bareiss (1969). The Levinson-Durbin recursion is a procedure in linear algebra to recursively calculate the solution to an equation involving a Toeplitz matrix, see Bäckström (2004).

13.13 Exercises

Exercise 1 Simulate a sample of size N that is from a mixture of distributions F_i , $i = 1$ to n , with probabilities p_i ($\sum_i p_i = 1$) and generate N random numbers from a categorical distribution of size n and probabilities p_i for $i = 1$ to n .

Exercise 2 Compute the covariance of two independent variables \mathbf{X} and \mathbf{Y} .

Exercise 3 Give example where Toeplitz matrices are useful.

References

- [Maple1987] Marple, L. S. (1987a) A fast computational algorithm for the modified covariance method of linear prediction. 1 of 3. Digital signal processing, Academic press, Academic Press.
- [Maple1987] Marple, L.S. (1987b). Digital Spectral Analysis with Application. Prentice Hall, NJ, USA. Oxford Dictionary of Statistics (2002), Oxford University Press.
- [Rodgers1988] Rodgers, J.L., Nicewander, W.A. (1988). Thirteen ways to look at the correlation coefficient. The American Statistician, 42.
- [Dempster1977] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B 39

- [Box1994]** Box, G. E. P. Jenkins, G. M.; Reinsel, G. C. (1994). Time Series Analysis: Forecasting and Control (3rd ed.). Upper Saddle River, NJ: Prentice-Hall.
- [Dunn2005]** Dunn, P.F. (2005). Measurement and Data Analysis for Engineering and Science. McGraw-Hill, New York.
- [Bareiss1969]** Bareiss, E.H. (1969), Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices, *Numerische Mathematik*, 13.
- [Baekstroem2004]** Bäckström, T. (2004). 2.2. Levinson-Durbin Recursion. Linear Predictive Modelling of Speech - Constraints and Line Spectrum Pair Decomposition. Doctoral thesis. Report no. 71 / Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing. Espoo, Finland.
- [Bilmes1998]** Bilmes J.A. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Techreport, April 1998.
- [McLaughlan2000]** McLaughlan, G.J. (2000), Finite Mixture Models, Wiley.
- [Hazas1999]** Hazas M. (1999), Localising speech, footsteps and other sounds using resource-constrained devices, IEEE, Xplore, Chicago, IL, USA, April, 2011.

Chapter 14

Fuzzy Logic and Rough Sets



Overview

As we have seen, there are several aspects of signal processing that do not have a true-false character but rather are subject to nuances of interpretation. This does not only concern the individual signals but the properties of signal processes. For their description, specific concepts describing uncertainty are needed. A very general concept is provided by fuzzy logic which has a formal description. This closes some gap between every-day conversation and formalism. Rough sets deal with decision made in the presence of uncertainty: Which decisions may still be valid and when does one need more information? This question is what we discuss first.

14.1 Rough Sets

Rough sets provide a useful method for dealing with uncertainty when one wants to make a decision. They are based on the observation that not everything is uncertain. It concerns when precise decisions have to be made despite uncertainty. It happens in everyday life regularly. Rough sets are equipped with some uncertainty but have also aspects of certainty, combining both. This is expressed in the basic definition. Suppose a set P is given as a subset of some universal set U that is uncertain.

Definition

(a) The *lower approximation* of P is

$$P_l = \{x \in U \mid \text{for all } y \text{ with } x \approx y : y \in P\}$$

(b) The *upper approximation* of P is

$$P_u = \{x \in U \mid \text{exists } y \text{ with } x \approx y : y \in P\}$$

The elements of P_l are surely in P and elements of P_u are possibly in P . The elements not in P_u are not in P . The set P_u/P_l is called the uncertainty area. The underlying idea is to notify the user that additional attention has to be paid here, for instance to obtain more information about the situation. However, often one has to make a decision in such a situation. For similarity measures we introduce rough sets. Suppose we have to make a decision based on acceptance of something. We look at the regions and we subdivide the uncertainty region further. This is the basis of the “rough” method to obtain some decisions despite uncertainty. For the uncertainty area one needs further information about the similarity measure. This gives rise to a judgement of P . Suppose we would only accept elements of P for a certain purpose. The advantage of introducing weak reject and weak accept is additional information to the user who can possibly react accordingly. Because decisions about elements in P_l and elements not in P_u are certain the rough set method can be regarded as “to be on the safe side”. For elements in the uncertainty area one has to look for further information in order to make a decision. The goal is of course to make the uncertainty area as small as possible. One should observe here that the rough set method gives an honest answer: It tells you where one is in an uncertain situation. The situation occurs in many cases, tolerances, probabilities, similarities and others (Fig. 14.1). Often one removes the uncertainty area by “brute force”, for instance like in fuzzy theory. This is discussed in the next section. Let us assume we have $x \approx y$ and $x_1 \approx y_1$. What to do? Figure 14.2 presented below shows how rough sets give a general advice to behave when any kind of uncertainty is present.



Fig. 14.1 Decisions using rough sets

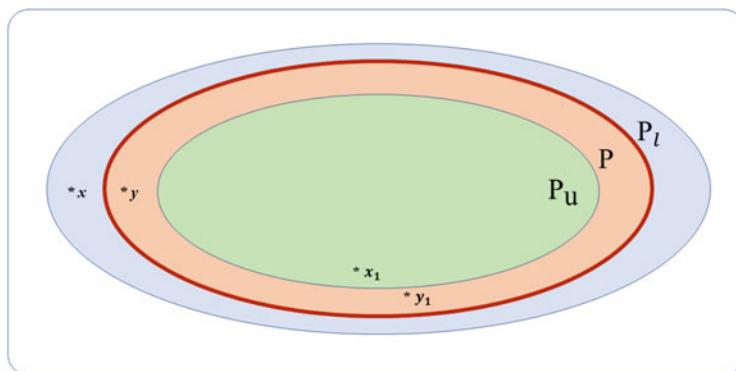


Fig. 14.2 Visualisation of rough sets

14.2 Fuzzy Sets

Fuzzy rule-based reasoning is a quite successful (approximate) reasoning methodology. Fuzzy reasoning can be regarded as an integration of logic and approximation. This is done with the cost that the classical logic view is somehow no longer applicable: The result of a rule application is no longer a traditional predicate. For this purpose a transformation into a 0–1 logic in order to make decisions is necessary but not always strictly possible. This operation is called defuzzification. One possibility to perform this is to make use of the rough set method. Fuzzy sets formalize the idea of graded membership and graded truth, i.e., the idea that an element can belong “more or less” to a set. A typical example is if one wants to express that something looks nice. Consequently, a fuzzy set can have “non-sharp” boundaries. In the same way a subset of a set is not sharp, elements do not belong precisely to a set, i.e. they are only more or less in a set. This gives some information about the elements, it has been extended to arbitrary properties. Although this may not be offering precision for a decision it may give help for a decision. Fuzzy reasoning tries to bring this to a computer. An example for signals is “the heartbeat is more or less normal”.

14.2.1 Basis Elements

The fuzzy approach deals with uncertain properties in a formal way. In everyday language they are quite common as informal expressions. For instance we may say “this set is quite large” or “the blood pressure is high”. In the context of signal processes they often occur; it is in the nature of the fact when we deal with stochastic processes. First we deal with some basic principle aspects that are simple to understand. Fuzzy membership functions assign values to elements of an arbitrary domain U with respect to a certain property where one does not accept just “yes” or “no” assignments:

$$\mu : U \rightarrow [0, 1]$$

Membership functions μ are associated with fuzzy subsets (or fuzzy predicates) P of U ; the membership reference to P is denoted by μ_P . $\mu_P(a)$ is called the degree with which a has the property P . Degrees occur frequently in applications concerning signals. It is a knowledge acquisition task to determine the degrees. It is obvious that these degrees depend on the context.

Example Look at the fuzzy predicate “large”. The definition is quite different if one considers “cars” or “cakes”.

Fuzzy reasoning has so-called linguistic rules. These are of the familiar form between expressions of any (for instance natural language) expressions:

$$A_1 \wedge A_2 \wedge \cdots \wedge A_n \rightarrow B$$

These expressions formulate opinions of experts how they would proceed. For instance if a car is large our parking place is insufficient.

In this example we see logical connectives. This is no accident. In fuzzy reasoning one uses all connectives known from logic and nothing else. Only the truth values are different and as a consequence the reasoning is too. Therefore each formula in fuzzy reasoning could verbally be interpreted as a logical formula. For computational purposes they are insufficient because of the lack of truth values. Therefore each expression is replaced by the actual value of the corresponding membership function. Hence the main difference to classical logic is in the truth values. For many purposes one can assume that U is totally ordered and takes for simplicity a real interval for U .

Because the expressions are combined with logical connectives we need some rules that have an analogy to the rules of logical connectives. The rules cannot be the same as in logic. There are two ways to combine different fuzzy sets: Intuitive and formal.

- (1) Composition operators are mainly *t-norms* and *co-t-norms* (corresponding to conjunction and disjunction) or the different kinds of implications. With the implications one can describe conditional fuzzy degrees, as compared to conditional probabilities. These operators define a local-global principle for fuzzy membership functions defined on complex objects.
- (2) The task of the next definition is to allow the same kind of reasoning that looks as in logic or in informal discussions. Even when uncertainty is presented one uses statements that borrowed something from the ordinary classical way. The point is how to get the fuzzy values for the combined formula formal logic although they do not have a precise meaning.

The intention is to incorporate the uncertainty. They the uncertainty area of the rough set method could be described. One way to look at it is to see it as an extension of probability theory. There are several possibilities for it. Hence one gets more than one choice for the analog of a classical notion. The intention is the same as in formal logic. Here the basic notions of combining formulas are introduced. There are many examples for explicit formulas but they are not presented here. Instead we will provide conditions that such combinations should satisfy; they are called axioms.

Definition

- (i) t-norms $f(x, y)$ (intended to compute $\mu(A \text{ and } B)$)

Axioms:

- (T1) $f(x, y) = f(y, x)$
- (T2) $f(x, f(y, z)) = f(f(x, y), z)$
- (T3) $x \leq x', y \leq y' \Rightarrow f(x, y) \leq f(x', y')$
- (T4) $f(x, 1) = x$

Typical t-norms are $f(x, y) = \min(x, y)$ or $f(x, y) = x \times y$.

- (ii) co-t-norms $f(x, y)$ (intended to compute $\mu(A \text{ and } B)$)

Axioms:

(T1), (T2) (T3) and

$$f(x, 0) = x$$

Typical co-t-norms are $f(x, y) = \max(x, y)$ or $f(x, y) = x + y - x \times y$

When one has defined any norms and co-norms satisfying the axioms one can start reasoning with fuzzy sets and one obtains the fuzzy values for the reasoning results.

14.2.2 Possibility and Necessity

An important part of fuzzy theory is possibility theory that deals with “degrees of possibility”. The term “possibility” is hence employed as a graded notion of possibly. A leading idea is to use the notion possible as a constraint. It is a kind of weak constraint but in a graded way. In some sense possibility theory deals with the basic informational principle underlying the possibility approach to knowledge representation and reasoning is stated as a principle of minimal specificity. That means, it excludes only elements where clear restrictions are known. In order to avoid any unjustified conclusions, one should represent a knowledge unit K by the largest possibility measure among those measures compatible with. Possibility measures are defined on subsets of the universe U under consideration.

Definition A mapping $[X, Y] : 2^U[0, 1]$ is a *possibility measure* if:

- (i) $\Pi(U) = 1$
- (ii) For all $X, Y \subseteq U$: $\Pi(X \cup Y) = \max(\Pi(X), \Pi(Y))$

Between possibility and certainty the necessity measures X, Y are located. They are defined by Eq. 14.1:

$$U(X) = 1 - \Pi(X) \tag{14.1}$$

where U is the universe. An event X is necessary in so far as its complement is not possible. It should be remarked that for some X both, $N(X)$ and $-\Pi(X)$ may be non-zero. In fact this is quite natural in the same way as to some degree the membership of an object to belonging to a set and to the complement is both possible.

14.3 Fuzzy Clustering

A fuzzy set U was defined as a mapping $\mu : U \rightarrow [0, 1]$ instead of a membership function. The situation is now that the sets are fuzzy sets and the clustering has to respect this. The main new issue is that an object can belong to more than one cluster. If clusters are fuzzy sets then an object can be in many clusters to some degree. **Fuzzy c-means** (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. The objective function is to minimize the number computed in Eq. 14.2 presented below.

$$J_m = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \|x_i - c_j\|^2 \quad (14.2)$$

With the following:

- $m > 1$
- μ_{ij} is the degree of membership of x_i in cluster j
- x_i is the i 'th data of d -dimensional measured data.
- c_j is the center of the j data cluster.

Fuzzy clustering is performed through an iterative optimization of the objective function shown in Eq. 14.2 with the update of membership μ_{ij} in Eq. 14.3 and the cluster centers c_j by iteration in Eq. 14.4:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_k\|}{\|x_i - c_j\|} \right)^{\frac{2}{m-1}}} \quad (14.3)$$

$$c_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m} \quad (14.4)$$

The user can determine a termination criterion.

Example We consider a one-dimension example in Fig. 14.3.

The data are represented on an X axis. Suppose the two clusters A and B are observed. The k-means algorithms would sharply separate the two in example 14.4.

As can be seen from the presented Fig. 14.4, the sets are sharply separated. Now assume that A and B are fuzzy sets. Fuzzy C-Means softens this curve (Fig. 14.5):



Fig. 14.3 Fuzzy set with one dimensional data

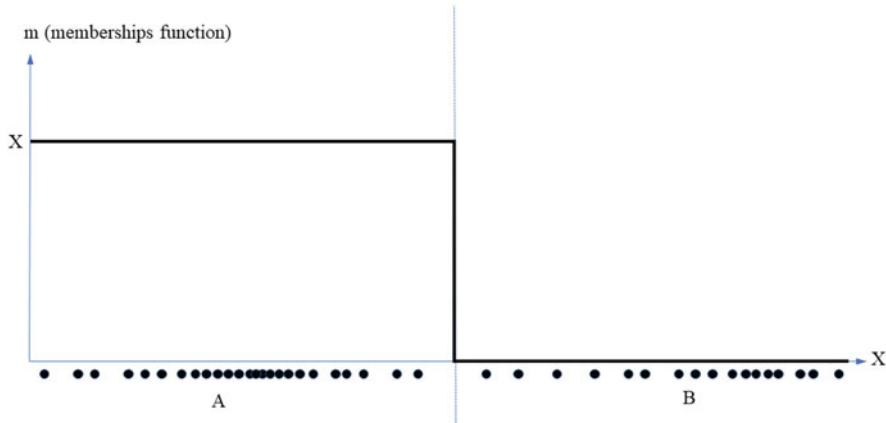


Fig. 14.4 Representation of strict membership

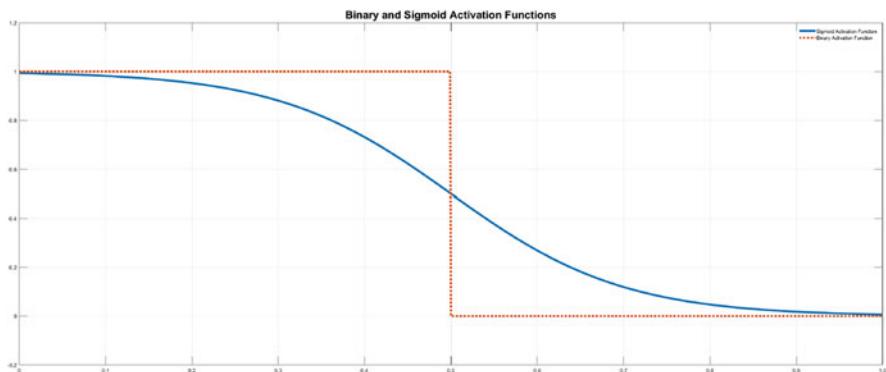


Fig. 14.5 Sigmoid membership function

We see that points are in a subset with different degrees of membership. The indicated point belongs to more than one cluster, but more to B than to A . This can support certain expectations or do the opposite. In this way one can combine it with EM.

14.4 Fuzzy Probabilities

They deal with events where the outcome is uncertain and only some probability is approximately known. The probability is a number but even this number may not be precisely known. For instance one may know that the next value of some number is over 0.5 with a high probability. In order to deal with such information one introduces the fuzzy value of probabilities. This is an aspect of hidden probabilities.

In fuzzy probability theory, we have an imprecision in our measurements, and random variables must be replaced by fuzzy random variables and events by fuzzy events. The probabilistic functions are replaced by their fuzzy analogs.

14.5 Background Information

General and Past

Bayesian reasoning is one of the most popular approach to deal with dynamically changing situations. More on Bayesian networks can be found in Jensen (1996). Rough sets have been introduced by Pawlak and Zdzisław (1982). See also Pawlak (1991). Fuzzy logic is a form of many-valued logic that is approximate rather exact. L. Zadeh has invented and popularized fuzzy sets and reasoning.

Suggestion

For an introduction to fuzzy sets a good text book is Lee (2005). A special aspect is when time is fuzzy represented. Estimation of temporal fuzzy sets that model dynamic processes have been used. Fuzzy reasoning is a way to deal with subjective judgments. Special attention of fuzzy sets to signals has been paid in Zimmermann (2001). This book gives also a comprehensive overview of fuzzy sets from theory to applications. Fuzzy probabilities are discussed extensively in Buckley (2005). An early work is in Zadeh (1984). There is a relation to subjective probability. In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters. See e.g. Bezdek and James (1981).

For more information, readers are suggested to read references used in this chapter (Kluwer Tanaka 2007; Giovanni et al. 2015) for more information.

14.6 Exercises

Exercise 1 Formulate the uncertainty area for fever degrees of a person and medical decision making using rough sets.

Exercise 2 Decide the acceptance of papers at a conference on the basis of rough sets.

References

- [Jensen1996]** Jensen, F.W. (1996): An Introduction to Bayesian Networks. UCL Press, London
- [Pawlak1982]** Pawlak, Zdzisław (1982). Rough sets. International Journal of Parallel Programming 11 (5): 341–356.
- [Pawlak1991]** Pawlak, Z (1991): Rough sets: Theoretical aspects of reasoning about data.
- [Tanaka2007]** Kluwer Tanaka K (2007). An Introduction to Fuzzy Logic for Practical Applications, Springer Verlag.
- [Lee2005]** K.H. Lee (2005): A first course in fuzzy theory and applications Springer Verlag.
- [Zimmermann2001]** Zimmermann, H.J. (2001) Fuzzy set theory—and its applications. Springer Science and Business Media, LLC, 2005
- [Buckley2005]** Buckley, J.J. (2005) Fuzzy Probabilities. Springer Verlag.
- [Zadeh1948]** Zadeh, L. (1984). Fuzzy Probabilities. Information Processing & Management 20, No. 3.
- [Bezdek1981]** Bezdek, James C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms.
- [Giovanni2015]** D'Urso, P., De Giovanni, L., Massari, R. (2015). Trimmed Fuzzy Clustering. Advances in Data Analysis and Classification.

Chapter 15

Neural Networks



Overview

Neural networks are computing systems with interconnected nodes that work much like neurons of the human brain. Neural networks are simple to formulate and widely used for computations. There are a wide variety of possible realizations. They are useful for the computations of properties and feature extractions of signal processes. They can recognize hidden patterns and correlations in raw data and classify it, and continuously learning and improving over time.

The first neural network was conceived of by McCulloch and Pitts (1943). They wrote a seminal paper on how neurons may work and modeled their ideas by creating a simple neural network using electrical circuits. This breakthrough model paved the way for neural network research in two radically different areas:

- (i) Simulation of how the human brain processes the information (biology).
- (ii) The use of neural networks in machine learning (computers).

The goal of the neural network research was to create a computational system that could solve problems just like a human brain would. However, over time, researchers shifted their focus to using neural networks to match specific tasks, leading to deviations from a strictly biological approaches. Since then, neural networks have supported diverse tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, etc.

As structured and unstructured data sizes increased to big data levels, people developed “deep learning” systems, which are essentially neural networks with many layers. Deep learning enables the capture and mining of more and bigger data, including unstructured data.

However, we will not give a full representation of the whole subject. Instead we will restrict ourselves to what is useful in this context of the book in order to understand the presented text.

15.1 Neural Network Types

Neural networks, also known as artificial neural networks (*ANNs*) or simulated neural networks (*SNNs*), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. There are many different kinds of neural networks—and each has advantages and disadvantages, depending upon the use.

Artificial neural networks are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer as depicted in Fig. 15.1. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Typical ANN structure is depicted in Fig. 15.1 below.

Examples of types of *ANN*'s include:

- **Convolutional neural networks (*CNNs*)** contain five or more types of layers, including: input, convolution, pooling, fully connected and output. Each layer has a specific purpose, like summarizing, connecting or activating. These are frequently used with deep networks composed of many layers, where the perceptrons in each layer are connected to a relatively small number of nodes in the previous layer, which forms its activation area. Convolutional neural networks have popularized image classification and object detection. However, *CNNs* have also been applied to other areas, such as natural language processing and forecasting.

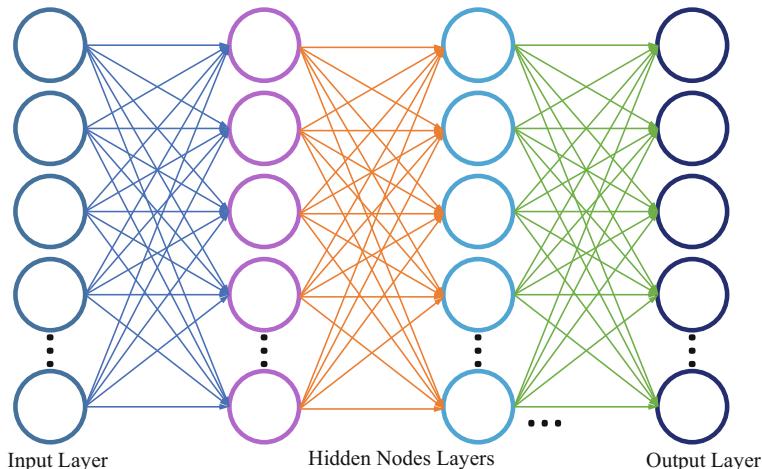


Fig. 15.1 An example of artificial neural network

- **Recurrent neural networks** (*RNNs*) use sequential information such as time-stamped data from a sensor device or a spoken sentence, composed of a sequence of terms. Unlike traditional neural networks, all inputs to a recurrent neural network are not independent of each other, and the output for each element depends on the computations of its preceding elements. *RNNs* are used in forecasting and time series applications, sentiment analysis and other text applications.
- **Feedforward neural networks**, in which each perceptron in one layer is connected to every perceptron from the next layer. Information is fed forward from one layer to the next in the forward direction only. There are no feedback loops.
- **Autoencoder neural networks** are used to create abstractions called encoders, created from a given set of inputs. Although similar to more traditional neural networks, autoencoders seek to model the inputs themselves, and therefore the method is considered unsupervised. The premise of autoencoders is to desensitize the irrelevant and sensitize the relevant. As layers are added, further abstractions are formulated at higher layers (layers closest to the point at which a decoder layer is introduced). These abstractions can then be used by linear or nonlinear classifiers.

Those various *ANN*'s provide only a sliver of configurations of types of *ANN*'s deployed with various learning algorithms. Next we will focus on learning algorithms for *ANN*.

15.1.1 Neural Network Training

There are two basic approaches that a neural network can be trained:

- Supervised Learning *ANN*
- Unsupervised Learning *ANN*

15.1.1.1 Supervised Learning

Supervised training requires a pair of input and corresponding output to be known. The output can be arbitrarily set to any desired value. However, it has to be noted, that the input-output relationship should be consistently applied in order for input-output mapping to converge as fast as possible. To apply the data, corresponding outputs are required to be included so that the network can find input-output mapping. This is done by defining the output labels for each input. As input data is fed into the *ANN*, it adjusts its weights until they are adjusted appropriately. In order to obtain the best possible model (e.g., adjusting the network weights) one has to be aware of the following:

- (i) Data preparation. Data has to be collected, and normalized.
 - (ii) Data has to be separated into:
 - (a) training set (80% of total data),
 - (b) validation set (10% of total data), and
 - (c) test set (10% of total data).
1. Development test set (50% of test data or 5% of total data)
 2. Evaluation test set (50% of test data or 5% of total data)

The typical supervised learning is applied in feed forward network with back-propagation training.

15.1.1.2 Unsupervised Learning

In unsupervised learning network no output is needed or specified. From the input data, the patterns are discovered. This is particularly useful when we are unsure of common properties within a data set. Learning algorithm are based on clustering. Common clustering algorithms are hierarchical, k-means, Gaussian mixture models, or self-organizing feature maps. Unsupervised machine learning algorithms are used to group unstructured data according to its similarities within the data-set. For example the various images of application of Self-Organizing Maps can be viewed by following this link: <https://www.superdatascience.com/blogs/self-organizing-maps-soms-how-do-self-organizing-maps-work/>.

15.1.2 Neural Network Topology

The arrangement of the nodes in an *ANN* is called its topology. First we restrict on the following topology of the network:

- (i) The nodes are organized in levels (as in example Fig. 15.1)
- (ii) From each node there is one connection/map to each node on the next level.

This is a very limited topology but it suffices to explain most major aspects of interest, although neural networks can be much more complex.

We distinguish the levels:

- (i) Input level (**Receptors**)
- (ii) Number of Hidden levels (**Hidden Layers**)
- (iii) Output level (**Effectors**)

The nodes on the lowest level are the input nodes and the nodes on the highest level are the effectors, i.e. result or output nodes. The input nodes are initially equipped with values, and values on the other nodes are computed by the maps. These are called feed-forward networks. This gives a mapping from the input nodes

to the output nodes. However, this ordering can be reversed. One can start with the output nodes and reversing the mappings until one reaches the input nodes. This is useful in order to see what is the reason for a result is. This process is called backpropagation. One starts with the results and goes backwards until input layer is reached. In this regard this procure can be considered supervised learning.

15.1.2.1 Beginnings

The history of artificial neural networks began with McCulloch and Pitts (1943). In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper on how neurons might work. In order to describe how neurons in the brain might work, they modeled a simple neural network using electrical circuits. Their work paved the way to parent state of development of *ANN*. The original McCulloch-Pitts model neuron contained the following:

- (i) $m > 0$ inputs x_1, x_2, \dots, x_m with Boolean values from $B = \{0, 1\}$
- (ii) One output value y containing a Boolean value.
- (iii) m weights w_1, w_2, \dots, w_m , (real numbers) associated to the input lines. Here the weights w_n are:
 - **excitatory** if $w_n > 0$;
 - **inhibitory** if $w_n < 0$;
 - $w_n = 0$ means “no connection”.
- (iv) Some threshold value θ for the output, also a real number.

15.1.2.2 Present State

Neural networks are built after the human brain. Hence, we summarize the terminology used.

- (i) Maps are called synapses
- (ii) The connected two neurons are called axon and dendrite

From Fig. 15.1 above, it can be observed that in order to make use of *ANN* one has to know the function of each individual node (e.g., node function) as well as how each node is interconnected (interconnected weights). Each *ANN* node can be described by activation value and its function as described by Eqs. 15.1 and 15.2

$$a = \sum_{i=0}^M w_i x_i + \theta = w_1 x_1 + w_2 x_2 + \dots + w_M x_M + \theta \quad (15.1)$$

$$y = f(a) = f\left(\sum_{i=0}^M w_i x_i + \theta\right) \quad (15.2)$$

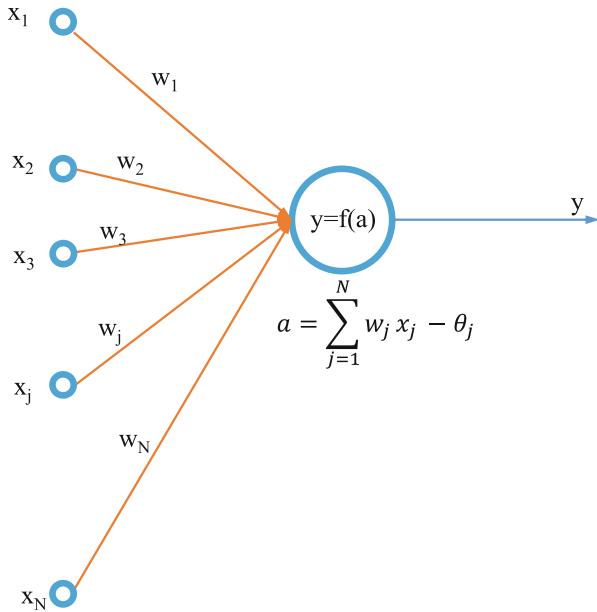


Fig. 15.2 An example of an artificial neuron

where f can be a multitude of activation functions; from binary threshold function to a more complex sigmoid functions as depicted in Eqs. 15.3 and 15.4 below, as well as their graphical representation as depicted in Fig. 15.2:

$$y = f(a) \quad (15.3)$$

and activation value a is

$$a = \sum_{i=0}^M w_i x_i + \theta \quad (15.4)$$

Two typical examples of activation functions are given below in Eqs. 15.5 for binary activation and 15.6 example of sigmoid activation function:

$$y = \begin{cases} 1 & \text{if } \sum_{i=0}^M w_i x_i + \theta \geq 0 \\ 0 & \text{if } \sum_{i=0}^M w_i x_i + \theta < 0 \end{cases} \quad (15.5)$$

and,

$$y = f(a) = \frac{1}{1 + e^{-a}} \quad (15.6)$$

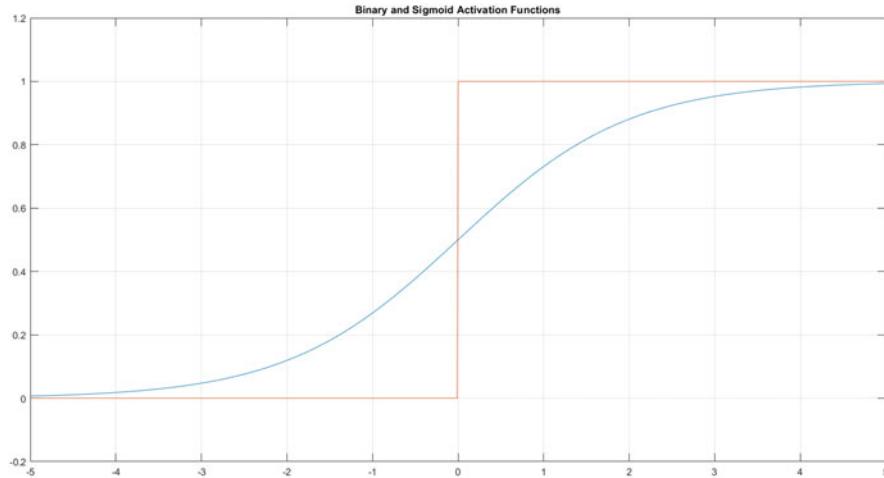


Fig. 15.3 Example of activation function utilized by a neuron

Note that the first neuron was proposed and modelled by McCulloch–Pitts in 1943, while the pattern classification ability of the *ANN* was developed latter (Fig. 15.3).

15.2 Parallel Distributed Processing

We will describe the use of Parallel Distributed Processing (PDP) first by explaining what kind of operation neurons perform (e.g., what is the function that they are performing as explained earlier through 15.1 and 15.2), as well as how they interconnected (e.g., topology of *ANN*). The following parameter are needed to define a complete neural network:

- N —number of processing units (e.g., neurons), $n \in N$.
- $a_n(t)$ —activation state of a neuron n at time t where $n \in N$,
- $a(t)$ —global network state at time t .
- $c_{m,n}(t)$ —connection states, denoting the local connection state between the neuron m and n .
- $c(t)$ —connection state of the network.
- $O_n(t) = f_n(a_n(t))$ —Output signal
- $O(t)$ —Output Vector
- $net_p(t) = g_n(c(t), O(t))$ —Input signal

In addition to those parameters we need to define two types of rules:

- Activation rule:

$$a(t+1) = f(t, a(t), \text{net}(t)) \quad (15.7)$$

- Learning rule:

$$D_t c_{m,n} = c_{m,n}(t+1) - c_{m,n}(t) = G(a_n(t), T_n(t)) \times H(O_n(t), c_{m,n}(t)) \quad (15.8)$$

- Teaching input: $T_n(t)$
- Input Vector: $T(t)$

At time t a neuron n sends an output signal $O_n(t)$ depending on its activation state, that is:

$$O_n(t) = f_n(a_n(t)) \in IR \quad (15.9)$$

The global output of the net is:

$$O(t) = (f_n(a_n(t)) | n \in N) \quad (15.10)$$

At time t a neuron m gets input signals from other neurons n . Net input is the part of the signals that arrives because of the inter-connectivity with other neurons. Local net input to neuron n at time t is

$$\text{net}_n(t) = g_n(c(t), O(t)) \quad (15.11)$$

Global net input into the network at time t is

$$\text{net}(t) = (\text{net}_n(t) | n \in N) \quad (15.12)$$

The organization of the nodes in some graphs is called the topology. The graph considered here organizes the nodes in levels. We look at maps that do not connect nodes on the same level but rather go from each node to all nodes on the next higher level. There is the input and the result level; all other levels are called hidden nodes. Figure 15.4 shows a feed forward net with one hidden level. This is a neural network where every node is connected to all nodes at the higher levels. No nodes on the same level are inter-connected.

15.2.1 **Forward and Backward Uses**

The net can be used in two directions:

- Forward: Given the input, what is the result?
- Fix the desired result and decide about the input nodes.

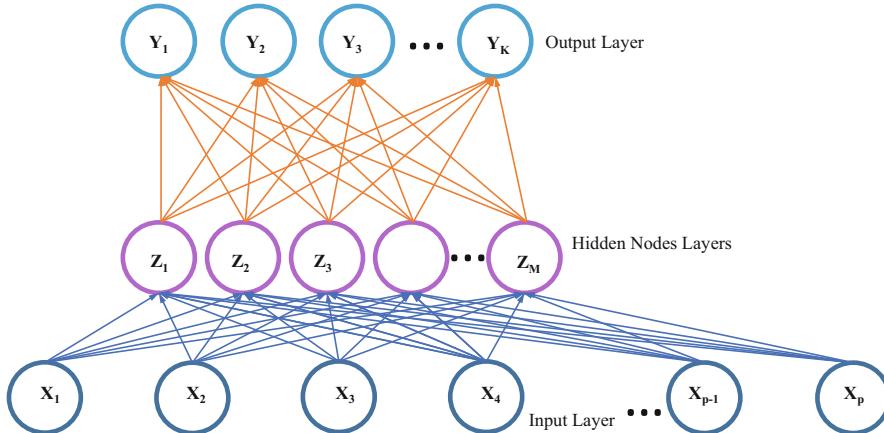


Fig. 15.4 An example of feed-forward neural network

15.2.2 Learning

All the parameters of the net are subject to be learned. Examples are:

- (i) Learn the connection state
- (ii) Learn the weights:

$$D_t c_{m,n} = c_{m,n}(t + 1) - c_{m,n}(t) \quad (15.13)$$

- (iii) Learn the input or the output

There are three possibilities:

- (i) Make $c_{m,n}(t + 1)$: Create a connection from m to n
- (ii) Remove $c_{m,n}(t + 1)$: Remove connection from m to n
- (iii) Modify $c_{m,n}(t + 1)$: Change connection from m to n

The leaning takes place if the computation does not lead to the desired result. This occurs in particular in optimization problems.

15.3 Applications to Signal Processing

For signal processes several neural network approaches have been used. Some are described in part C for biomedical, seismic areas and multimedia systems. The Neural Network is established in industry. The main applications are in situations where there are no rules or one does not know how to handle them efficiently. In more detail one has the following application types. First a classical problem

in signal processing is to extract each source signal without any knowledge either about those signals or about their combination in the sensors outputs. This problem is studied on neural networks where any message appears as an unknown mixing of primary entities. Other problems are of the prediction type. One has the result of signal processes and wants to know what the interesting information provided is. The approaches using neural nets are frequent in biosciences and in seismic applications. In these cases it may occur that learning for the neurons is necessary. An additional problem arises when continuous signals come in. Neural networks can transform these into discrete ones. Neural network algorithms have been proposed for solving this problem. The problem of blind signal separation arises in many areas such as speech recognition.

15.4 Background Information

Neural networks are quite popular today for many applications. Many of the techniques for neural networks have applications for dealing with signal processes. Some are discussed above. These techniques are very diverse.

In particular we concentrate on McCulloch neurons and feed forward nets using backpropagation. This restricts the topology of the nets as well as the nature of the neurons. An early publication is McCulloch and Pitts (1943). A book with many contributions is Yu et al. (2001). It contains articles of experts about their own domain in signal processing. For optimization see Cichocki and Unbehauen (1993). Neural nets are in particular used for signal prediction. The predicted values are on the top nodes of the networks. The prediction of classically secreted proteins is treated in Bendtsen et al. (2004). An application with learning is in Kosko (1990). A tool for CBR is CBR Works. The topology of the net can still almost freely be chosen. The one here selected is called feed-forward nets.

Suggested

MATLAB provides several tools that utilize Neural Networks. Example of usage of neural networks can be found from their website. For example, one can perform Neural Network training with [Deep Learning Toolbox](#) by invoking `deepNetworkDesigner` command in the MATLAB window. The network is described in “Get Started with Deep Network Designer”. In addition to that tool, there is a Deep Neural Network speech recognition training procedure described from the website ‘Speech Command Recognition Using Deep Learning’. The given instructions explain building of the tool that recognizes commands spoken in free speech. This requires running a large script also provided from the MATLAB website. More information can be found from the link [Speech Command Recognition Using Deep Learning](#).

15.5 Exercises

Exercise 1 Look at the Boolean function XOR. Can you represent it by a single McCP neuron? Support your answer.

Exercise 2 Find a way to represent a complex Boolean function of your choice with neural nets.

Exercise 3 Define and implement a neural net that discovers a zero in a sequence of Boolean values.

References

- [**Yu2001**] Yu H.H., Jenq-Neng H. (Editors, 2001). *Handbook of Neural Network Signal Processing*. CRC Press
- [**Cochocki1993**] Cochocki, A. Unbehauen, R. (1993) *Neural Networks for Optimization and Signal Processing*. J. Wiley
- [**McCulloch1943**] McCulloch, W., and Pitts, W. (1943). A logical calculus of ideas imminent in nervous activity, *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133
- [**Kosko1990**] Kosko, B. (1990) Unsupervised learning in noise. *Neural Networks, IEEE Transactions on*, 1990. ieeexplore.ieee.org
- [**Bendtsen2004**] Bendtsen, J.D., Nielsen, H., van Heijne G. (2004). Improved prediction of signal peptides. *Journal of Molecular Biology* Vol. 427

Part III

Real Aspects and Applications

In Part III the reader is informed about a wide range of topics that occur in applications. It contains the required extensions or modifications of techniques that have been introduced in earlier sections.

Generalities About Part III

The reader of this part should be familiar with the sections in Parts I and II corresponding to the addressed techniques. This applies in particular to stochastic processes. In this part we extend Part II in various ways. The main view is how applications influence signal processes, how they and their properties look like and how useful they are. Techniques are refined in different ways depending on the application.

As it will be seen, each application type has its own characteristics. It depends mainly on two aspects:

- How signal processes are generated.
- Which information in the processes is of interest, how to produce and understand it.

A first issue is to get the signals. Sometimes this is problematic, for instance if the source is inside of another object like a human body. A central point is now that the signal processes contain information. This is obvious for speech but it occurs in every signal process. In speech, there is the topic of psychoacoustics phenomena that expresses information beyond the pure literal meaning. That happens in other signal processes too. The applications require different methods of processing signals that will be discussed individually in the text. They are in the spirit of psychoacoustics but the methods go further. In addition, the applications take place in environments of different characters that one has to respect. In areas like medicine the reply is most important: The two machines exchange information several times and therefore between these machines a communication has to take place. The signals in this

chapter have different characteristics in order to cover as many aspects as possible. For the seismic signals, the location of the source is important.

Overview of Part III

A part of reality is the environment in which the signals occur. The environment creates noise. The noise splits up into different types, which require different treatments. Difficult situations arise if mixed and complex environments occur. They result in mixed kinds of noise that take place simultaneously and create special problems. In each application a signal and in particular a set of signals has a meaning. Part of the meaning is for humans the expectation of an intended reply of the signal to the receiving entity. This is hidden, however. Often a reply is expected but in many situations there is no such expectation. This is mainly the case when the receiver is a machine.

Sample areas where signals belong are for instance linguistics and medicine. They are traditionally concerned with meaning and understanding. We rather rely more on syntactic elements. There are simpler methods not involving understanding. They work because we have a simpler situation too. Signals are not directly questions. But the humans who understand the meaning of the signal processes may react as if they were questions. Therefore, they provide a reply. If a machine does the reception, the situation is often that the signal process is a command and the expectation is that the command should be performed. Hence the reply can be an answer to a question or the execution of an intended action. One has to describe this action; the precise description depends on the application. In general, the reply needs an additional machine. Here we have two machines:

- One for accepting and understanding the signal process
- One for the reply.

Main Topics in Part III

An important part in applications is that they take place in some environment. There other activities take place that usually are not silent. Often, the corruption creates the existence of noise.

Therefore, one has to detect and describe the noise properly. For one description of noise strength we use the Speech-Noise-Relation (SNR). Then we consider as a central topic that environment corrupts the signal. Noise corrupts the signals before reactions can take place, and the meaning will differ from the wanted one. Hence, we want to remove or at least diminish some important types of noise.

Noise removal is a major part in the introduction of application discussions. The removal techniques differ for the introduced types of noise. Each type requires a

specific removal technique. A special challenge arises if such kinds of noise occur simultaneously. We consider this as an important issue. In Kalman filtering one has developed methods to draw conclusions about an actual state from only partially correct observations. This is one in an iterative way and a kind of learning.

Furthermore we discuss specific examples where digital processes occur. Speech occurs in different situations. A main part of the discussion is reserved to speech recognition because this is necessary for communication. For a successful recognition the study of the human speech generation process is useful.

Major applications of speech recognition are discussed too. As already said, the number of signals in a spoken word is too high for combinatorial investigation. In order to diminish these signal numbers one introduces features. These are short vectors that still contain relevant information about the speech. There are different kinds of features that are relevant for speech and different feature extraction methods. We will present several of them. In addition, we discuss psychoacoustic speech elements.

We consider the aspects of production, reception, and interpretation of speech in two different ways. One is from the human body view and the other one is from the automata view. Further some other important tasks of signal processing applications are considered. Next, we consider a selection of other areas where signals play an essential role. This includes first bio-medical applications. Here we consider medical issues namely ECG and EEG. Both are dominated by signal processes. Another area is the seismic domain. Here the signals come from the earth and are interpreted by humans. At the end of the section we compare different methods from this viewpoint. The comparison of the applications shows commonalities and difference of the methods.

Chapter 16

Noisy Signals



Overview

Noise can be considered as unwanted sound that is unpleasant, loud or disruptive to hearing. From a physics standpoint, noise is indistinguishable from desired sound, as both are vibrations through a medium. The difference arises when the brain receives and perceives a sound. Acoustic signal is any sound that is produced in the acoustic domain, either deliberate (e.g., music or speech) or unintended (e.g., noise). In general one tries to avoid noise. Noises are signals! They have undesired effect of “disturbing” any kind of original signal and hence are undesirable. The original processes, including noisy ones, where messages are send to humans or to other machines will be discussed. Handling signals in presence of noise is paramount where the methods have to be developed to extract original messages that are embedded in the noise. If this is not done, either implicitly or explicitly, the original messages may be misunderstood and the intended actions may not be performed correctly. An example is bio-medical signals where incorrect signals can ultimately lead to incorrect categorisation of the health condition of the patient.

16.1 Introduction

The general view of the reception of different physical properties of speech in noisy signals, sounds that occur within speech through varying frequencies, amplitudes and duration are shown in Figs. 16.1, 16.2, and 16.3. Preliminary analysis of signal waveform in time, frequency (pitch) and amplitude ('loudness' or intensity) of the sound ("Oeffne die Tuer": spoken in German, which literally means "open the door"). In the first row signal waveform in time domain is depicted. In the second row, the pitch depicting the frequency of fundamental frequency, that is pitch. And finally the third figure depicting the amplitude of a spectrum presented in the third row. The spectrum provides more complex information than the waveform. It shows

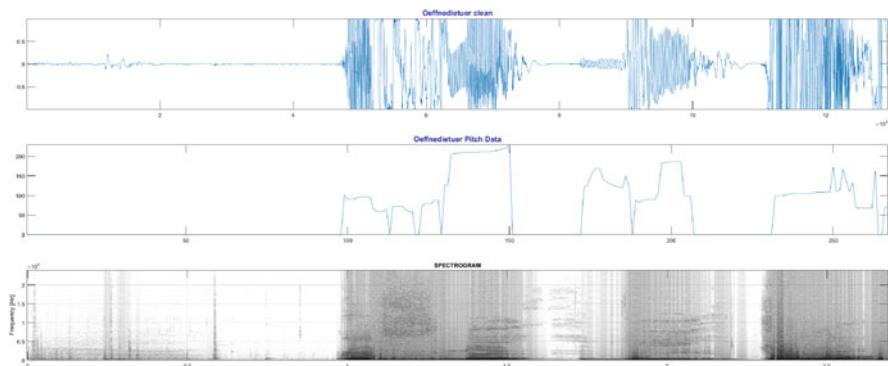


Fig. 16.1 Example of the waveform (e.g. speech Data), pitch (e.g., pitch data), and spectrographic display (e.g. spectrogram) of clean spoken utterance ‘oöffne die tuer’ in a standard/ quiet environment

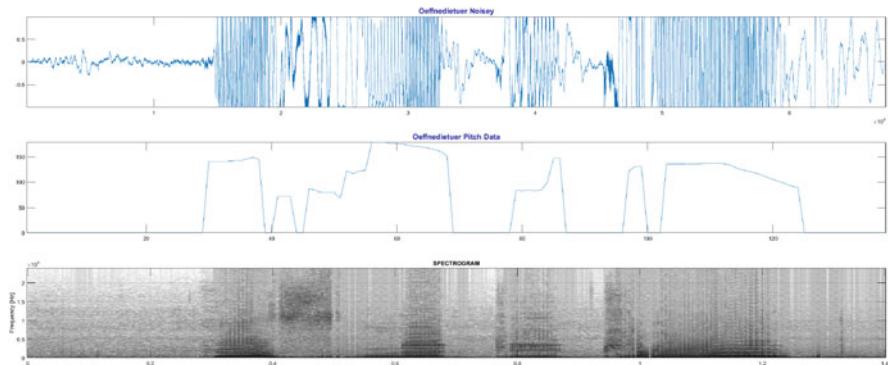


Fig. 16.2 Example of the waveform (e.g. speech data), pitch (e.g., pitch data), and spectrographic display (e.g. spectrogram) of speech ‘oöffne die tuer’ recorded in an noisy industrial environment. Various types of noises were present due to heavy duty of machinery and running motors providing noisy environment. These machines were in **resting state**

energy of the signal in time and frequency. The amplitude of the signal is represented in spectrum as the dark regions proportional to the acoustic energy. The louder the sound, the darker it appears on spectrographs and is therefore represents regions with more energy.

If we look closely to the recordings, we see that audio of the recorded files are being clipped. This is due to the fact that speaker was speaking to laud or recording was set to high, hence the signal that was recorded exceeded dynamic range of values afforded for usual recording. If waveforms are clipped they will give wrong results and thus should be avoided. For these cases, lowering the recording volume is necessary to avoid overloading (see Chap. 1 Digital Signal Representation).

Noise and the environment both have an influence on the result, because they commonly coincide and also because the environment can be the source of the noise.

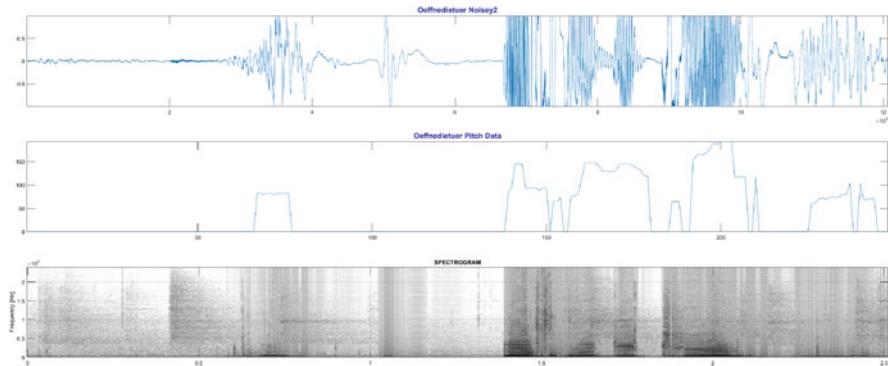


Fig. 16.3 Example of the waveform (e.g. speech data), pitch (e.g., pitch data), and spectrographic display (e.g. spectrogram) of speech ‘oeffnedietuer’ recorded in an noisy industrial environment. Various types of noises were present due to heavy duty of machinery and running motors providing noisy environment. These machines were in **running state** generating varying load sounds. Such industrial environment have always varying types of noisy sounds. In such environments, the noise can be so high that speaker can not distinguish the speech from noise

The general pattern of solving the noise issue lies in three categories:

- (i) transform signal by removing noise (e.g., by applying the noise compensation),
- (ii) designing signal features that are noise tolerant,
- (iii) train models with signals that contain noise, and
- (iv) transform models to match noisy features and compensate for their negative effects.

Removing noise can be done using the signal processing methods to clean the noisy signal. For example by having microphone array that utilize any number of microphones focusing the attention on a specific signal by the technique called Beam-forming as depicted in Fig. 16.4.

This technique is suited for removing the noise in a constrained environment like a room. It provides optimal control in capturing the signal by suppressing undesired sounds and noise by using multiple microphones to capture the speech signal. Microphone arrays record the speech signal simultaneously over a number of spatially separated channels. Many array-signal-processing techniques have been developed to combine the signals in the array to achieve a substantial improvement in the signal-to-noise ratio (SNR) of the processed signal.

The general view of a noisy signal is shown in Fig. 16.5 is presented below.

When it comes to different types of noise, we want to replicate how the human ear interprets noise in order to get an accurate representation of its impact. For example, electronic devices generate unwanted random addition to the signal that is considered to be a noise. Often, the signal has a complex pattern. In speech, the noise is mixed with different tones and typically varies with time (e.g., non stationary signal). This occurs generally in stochastic processes, mainly in the form of HSM. The complex tonal sound has more processing complexities as compared

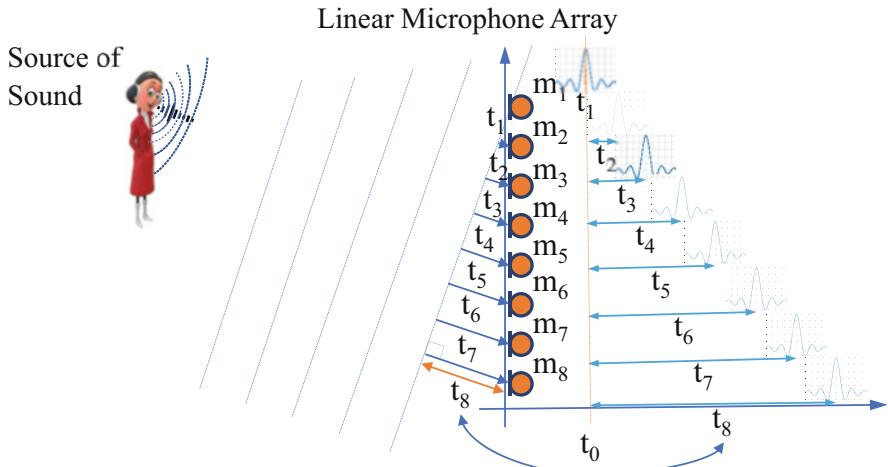


Fig. 16.4 Linear microphone array for signal processing in a typical noisy environment. Number of microphones m_i for $i = 1, \dots, 8$ capturing the signal. Arrival times t_i for $i = 1, \dots, 8$ of the signals in the array. Signal is captured at different times due to varying paths of arrivals of the wave pattern

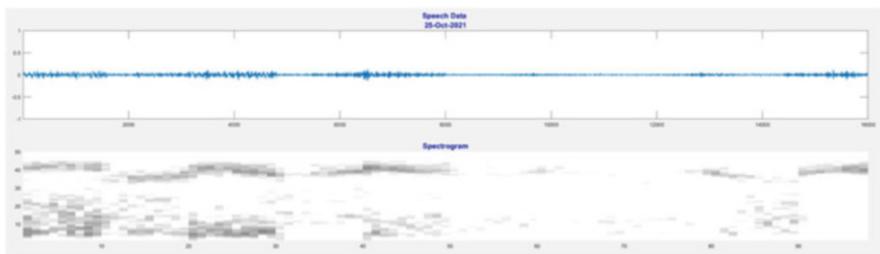


Fig. 16.5 Example of the Noisy Waveform and its Spectrographic display

to a pure sinusoidal tonal sound. The variability of the signal makes the research complicated and challenging task. A given message that is generated by a particular sound source repeated several times is not the same each time. This is due to the fact that the source often is not stationary. Addressing the noise issue one has to deal with the following:

- Identifying Noise,
- Noise Modeling,
- Developing Noise Reduction Techniques,

16.2 Noise Questions

Technically the basic questions connected with noise are:

- What are the sources of noise?
- Are there different types of noise?
- How loud is the noise?
- Does the noise disturb the reception of signals?
- In case of disturbance, what should one do?
- etc.

We will discuss these questions step by step. This will lead to a longer investigation of the nature of noise. Some types have already been mentioned.

16.3 Sources of Noise

Some important sources that create noises are listed below.

- (a) **Everyday noise:** created from moving or colliding sources and is the most familiar type of noise in everyday environments. In addition one has people talking in the background.
- (b) **Outside Noise:** it is general noise that is generated by moving cars, air-conditioners, computer fans, traffic, wind, rain, etc.
- (c) **Electromagnetic noise:** present at all frequencies and in particular at the radio frequencies. All electric devices, such as radio and television transmitters and receivers, generate electromagnetic noise.
- (d) **Electrostatic noise:** generated by the presence of a voltage with or without current flow. Fluorescent lighting is one of the more common sources of electrostatic noise.
- (e) **Signal noise:** the noise that results from the digital/analog processing of signals, e.g. quantization noise in digital coding of speech or image signals, packets in digital data communication systems.

A first step for detecting the nature of noise is to measure it. For this one needs to know the location of the source of the noise. Thus, here is a first question that one has to answer.

16.4 Noise Measurement

The ambient noise is regarded as taking place in a bounded space. We rely on our perception process based on the loudness of the sound in the environment.

Below we will introduce some technical concepts such as the probability density function (pdf) or the box plot to learn the underlying distributions of the data. We focus on realizing the energy trend of the noisy speech, the distribution of data i.e. how the variables map with respect to shape, variance and the probability of the range of the data. In the box plot, we try to know the extremity of the frequencies with respect to median position. We rely on the sound level measurement using A-weighting filters that approximates the human ear perception. Our goal is to develop a speech recognition system as presented in examples in following chapters. In addition, this filter is standard in the ambient environmental sound level measurement. A-weighting filter measures the loudness as the average sound level over the period as a root mean squared power in dB. According to statistical information and the sound level measurement, we have continuous, varying, intermittent, and impulse types of sound. And in the analysis provided by our measurements we have random noise level variation such as mixed noise levels varying from 60 dB to 115 dB where communication is barely audible, loud or dominant noise levels are over the environment where the communication is not even possible among the human beings. In such measurements, the range of 70–80 dB is in the mild-steady noise level, 80–89 dB is in the varying steady-unsteady and above 90 dB is strong noise level where the loudness of the sound is extreme. The real situation does not always satisfy this condition and therefore we take the enhanced signal as the system output which is clearly audible. The measurement of the noisy signal characteristics and the solution to the noisy signal problem can be applied utilizing methods that are passive, or active, or a combination of both. Here we will introduce the active and passive actions and their applications.

- By “active” we mean actions that change something such as removal, filtering, statistical modeling, matched filtering operations, and Kalman filtering.
- By “passive” we mean some standard observations and measurements of the noisy speech or the noisy sounds that do not change anything.

Usually one takes the passive approaches first. They provide information in order to perform the actions properly.

Examples Suppose we are in a noisy room and want to communicate with persons. See also chapters describing many applications: Chaps. 23 and 24 where one can find many technical details. The main active method is noise removal. This can mean complete removal or reduction to a lower levels.

16.5 Weights and A-Weights

Weight functions are filters specific for speech. A weight function is used when performing a sum, an integral, or average to give some elements more “weight” or influence on the result than other elements in the same set. There several different weights named as A, B, C or D weights. For the purposes of this book the so-called

Table 16.1 A-weights in dB in frequency range from 63 to 2000 Hz

Frequency (Hz)	63	125	250	500	1000	2000
A-weights (dB)	-26.2	-26.1	-8.6	-3.2	0	+1.2

A-weights are of interest. Here we explain how one can use the filter to evaluate the signal level of the noisy signal. For this purpose one uses the A-weighting filter in the frequency domain. First the spectrum of the noisy signal is computed using the discrete Fourier transform (DFT) and the A weighting filter is then applied to measure the signal power in dB. The power spectrum S of the N sampled signals $s[n]$ where $n = 0, 1, 2, \dots, N - 1$ is computed by equation

$$S(k) = \sum_{n=0}^{N-1} |s[n]| e^{-\frac{2\pi kn}{N}} \quad (16.1)$$

The equation below shows a relation between the frequency response of the A-weighting filter denoted here by $\alpha_A(f)$ and the linear frequency of the signal denoted by f in Eq. 16.2:

$$\alpha_A(f) = \frac{12200^2 f^4}{(f^2 + 20.5998997^2)^2 (f^2 + 12200^2) (f^2 + 107^2)^{0.5} (f^2 + 737.9^2)^{0.5}} \quad (16.2)$$

This formula looks quite complicated. The result presented in the formula 16.2 was obtained by a large number of experiments. Hence, it is not the result of some theory. Please look at the Background Information section. Such reasoning occurs quite often.

An example of A-weighting is shown in Table 16.1 where the frequency ranges from 63 to 2 kHz and the A-weight increases with the frequency.

The A-weighting for varying signal level is visualized in Fig. 16.6 and for strong noise is shown in Fig. 16.7.

The A-weighted signal level measurement $S_A(k)$ is computed by the multiplication of the A-weighting filter frequency response and the noisy signal spectrum $S(k)$ in Eq. 16.3.

$$S_A(k) = \alpha_A(f_k) S(k) \quad (16.3)$$

The signal energy ζ shown in Eq. 16.4 is computed by squaring the spectrum of $S_A(k)$ for $k = 0, 1, \dots, \frac{N}{2} - 1$. That is:

$$\zeta = \sum_{k=0}^{\frac{N}{2}-1} S_A^2(k) \quad (16.4)$$

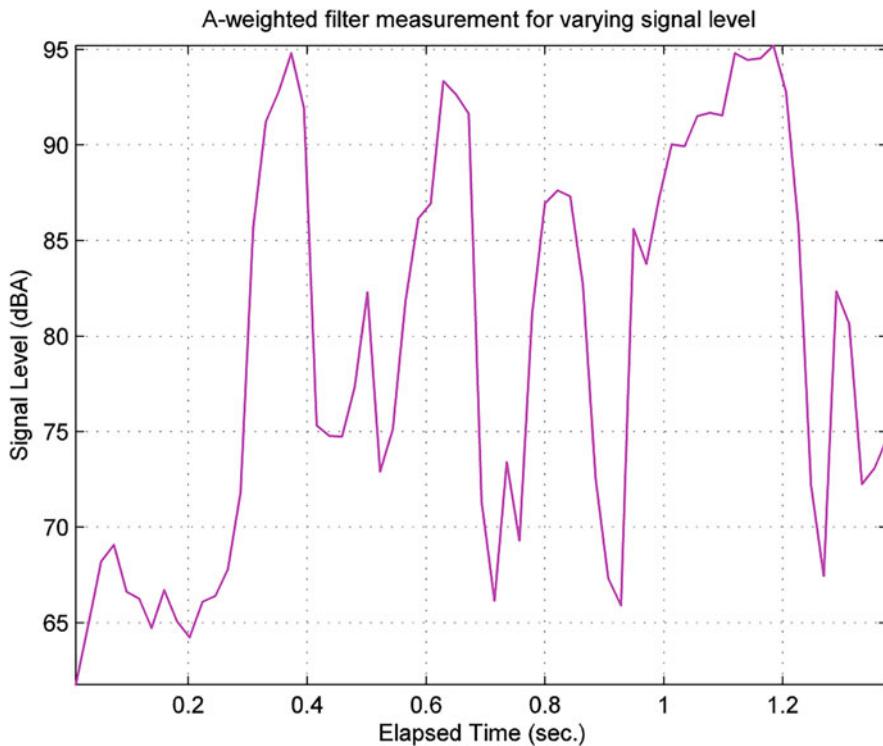


Fig. 16.6 Example of the A weighted filter measurement using varying noisy signal

The signal level in dBA is computed by Eq. 16.5, presented below, where ζ_{ref} is reference pressure and its value is 0:000204 dynes/cm². The signal level is in dBA:

$$\text{Signal level in DBA} = 10 \log_{10} \frac{\zeta}{\zeta_{ref}} = 10 \log_{10} \zeta - 10 \log_{10} \zeta_{ref} \quad (16.5)$$

In Fig. 16.8 we observe the various sound level, time and frequency information of the hybrid noisy signal. In Eq. 16.5, ζ_{ref} is a constant which is replaced by a

$$\text{Signal level in DBA} = 10 \log_{10} \zeta + C \quad (16.6)$$

In our measurements we use a calibration constant of 55 dBA because in a noisy environment a human being can perceive sounds in the range of 50–90 dBA.

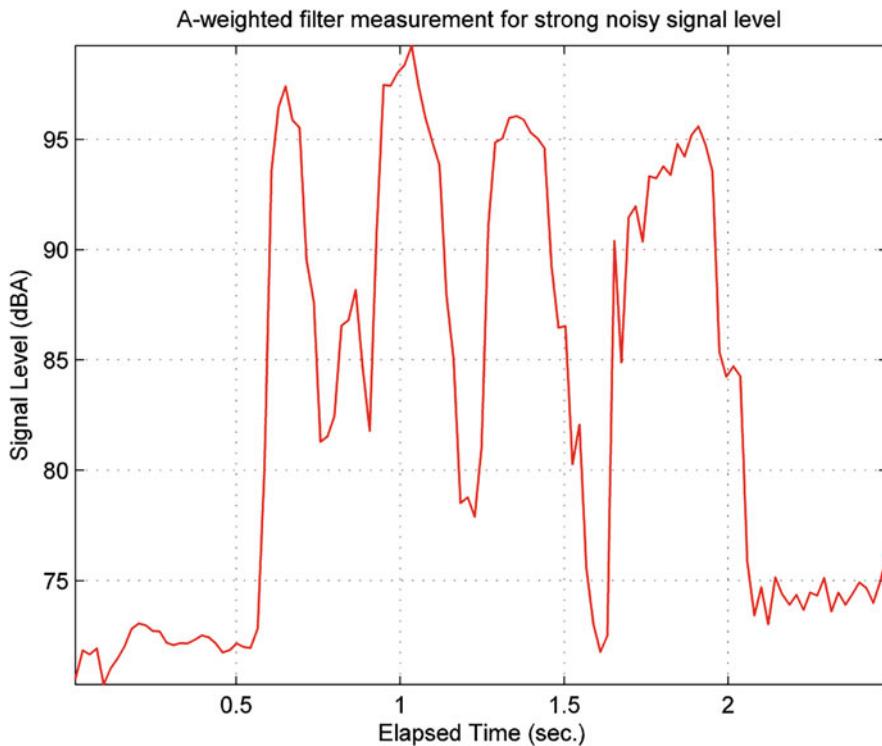


Fig. 16.7 Example of the A weighted filter measurement of the strong noisy signal

16.6 Signal to Noise Ratio (SNR)

When noise occurs in a signal process one wants to judge how influential the noise is. We judge the signal intelligibility mainly by our perception or by a machine perception. This subjective method is replaced by objective measurement of computing the [Signal to Noise Ratio](#) (SNR). The intelligibility measurements provides for categorization of the signal if it is audible or not. However, the signal strength and noise strength is primarily measured by the signal to noise ratio (SNR) in dB. This gives a relative performance of the signal with respect to noise. If the signal strength is higher than the noise strength, the ratio is positive. If the noise strength is higher than the signal strength, and then the SNR is negative. The signal strength and noise are displayed along with the signal to noise ratio (SNR) in dB for comparing relative performance. dB is a logarithmic scale for measuring relative performance or evaluates the signal usability as a function of SNR. For example, 3 dB corresponds to just a noticeable performance in signal strength from doubling the power. 10 dB is a significant improvement in signal strength from 10 times the power. With M is indicated the number of frames, with n the number of samples in

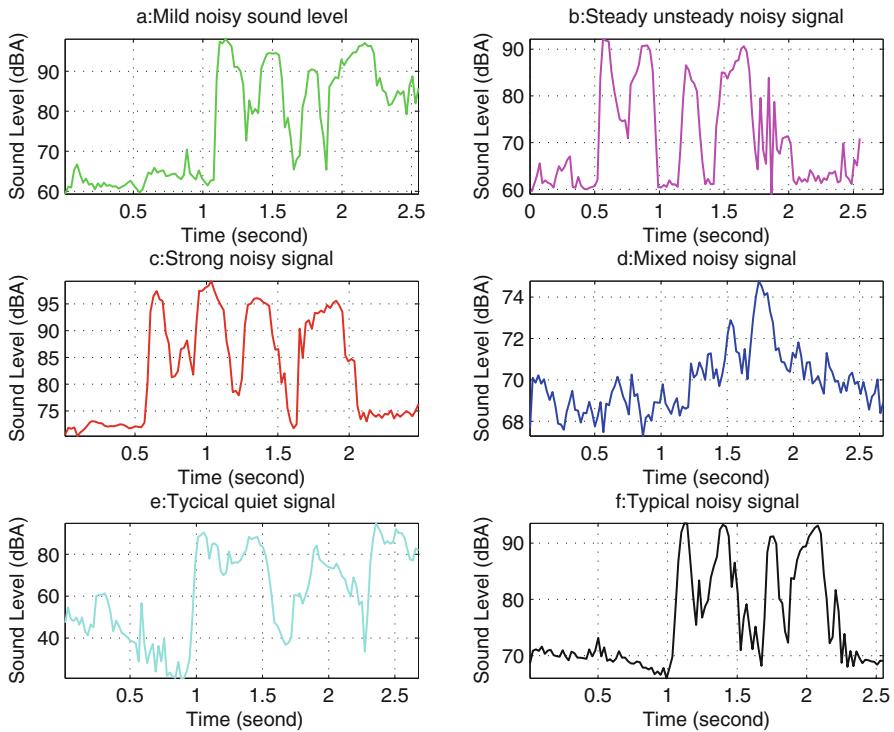


Fig. 16.8 The A weighted filter measurement of different noises

a frame is denoted, y_n is noisy speech, while with b_n is the noise collected from the environment absent of any speech. Furthermore, P_s indicates the power of the signal and P_n is the power of the noise. The whole signal is divided into M segments, as indicated below in Eq. 16.7, such that $m = 1, 2, \dots, M$. There N is the number of samples in each m . $y_m[n]$ is the noisy signal, $b_m[n]$ is the noise collected from the environment. The SNR is defined and computed by equation.

$$SNR_{\sigma} = \frac{1}{M} \sum_{M=0}^{M-1} \log_{10} \sum_{n=0}^{N-1} \frac{y_n}{y_n - b_n} \quad (16.7)$$

If the signal strength is higher than the noise strength, the ratio is positive, but if the ratio is negative or the noise strength is higher than the signal strength, than the SNR is negative. In such cases, reliable communication is not possible and the necessity of reduction of the noise in the desired signal becomes significant for an effective communication. Hence, the methods of handling noise are necessary.

Example We take a 20 ms rectangular pulse sampled for 2 s at 10 kHz. The current noise is a Gaussian noise. The data are:

- Rng (random number generator): default
- $T_{pulse} = 20\text{e-}3$;
- $F_s = 10\text{e}3$;
- $t = -1:1/F_s:1$;
- $x = \text{rectpuls}(t, T_{pulse})$;
- $y = 0.00001 * \text{randn}(\text{size}(x))$;
- $s = x + y$;
- $\text{pulseSNR} = \text{snr}(x, s - x)$, $\text{pulseSNR} = 80.0818$

16.7 Noise Measuring Filters and Evaluation

As mentioned, we regard our situation as ambient noise control because the situation is taking place in a bounded space. Initially we rely on our perception process based on the loudness of the sound in the environment. We used the basic concepts such as the probability density function (pdf) underlying the distributions of the data. We focus on realizing the energy trend of the noise and noisy speech, the distribution of data i.e. how the variables map with respect to shape, variance and the probability of the range of the data. We rely on the sound level measurement using A -weighting filter, among others weighted filters as B , C , D , E filters, because A -weighting filter approximates the human ear perception, and that feature alone makes it important. In addition, this filter is standard in the ambient environmental sound level measurement. A -weighting filter measures the loudness of the average sound level over the period as a root mean squared power in dB. According to statistical information and the sound level measurement, we have continuous, varying, intermittent, and impulse types of sound. Most measurements have random noise level variation.

16.8 Types of Noise

Noise types can be introduced from various points of views. One can, for instance, consider the sources or the physical structure. We look at them from the aspect of the impact of the receiver. This can be a human or another machine. If one formalizes the receiver one can use a machine for the reception. This is the aspect of concern in this book. In this respect we look at the four types of noise introduced above in some detail. We repeat:

1. Residual noise,
2. Mild noise,
3. Steady-unsteady time varying,
4. Strong noise.

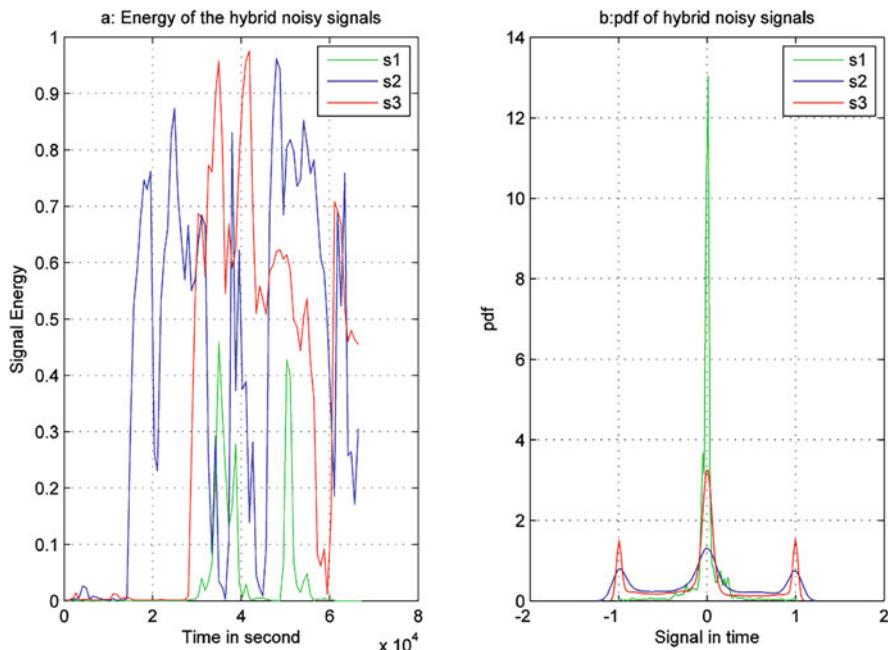


Fig. 16.9 Hybrid noise

First, we will give a short explanation of these terms before we go to the details and applications. Figure 16.9 shows the energy amplitudes of the different types. The energy for the different types of noises is shown in Fig. 16.9, the energy types is shown in Fig. 16.10, and the signal energy in a segmented or framed signal is shown in Fig. 16.11.

According to statistical information and the sound level measurement, we have continuous, varying, intermittent, and impulse types of sound. Our measurements have a random noise level variation such as mixed noise levels. They vary from 60 dB to 115 dB where communication is barely audible to loud—dominant noise levels over the environment where the communication is not even possible among the human beings. In such measurements, the range of 70–80 dB is in the mild-steady noise level, 80–89 dB is in the varying steady-unsteady, and above 90 dB in strong noise level where the loudness of the sound is extreme. Although we follow standard noise levels, in cases where the real signal is not audible, hence we take the enhanced version of the signal as the system output, which makes the enhanced signal audible.

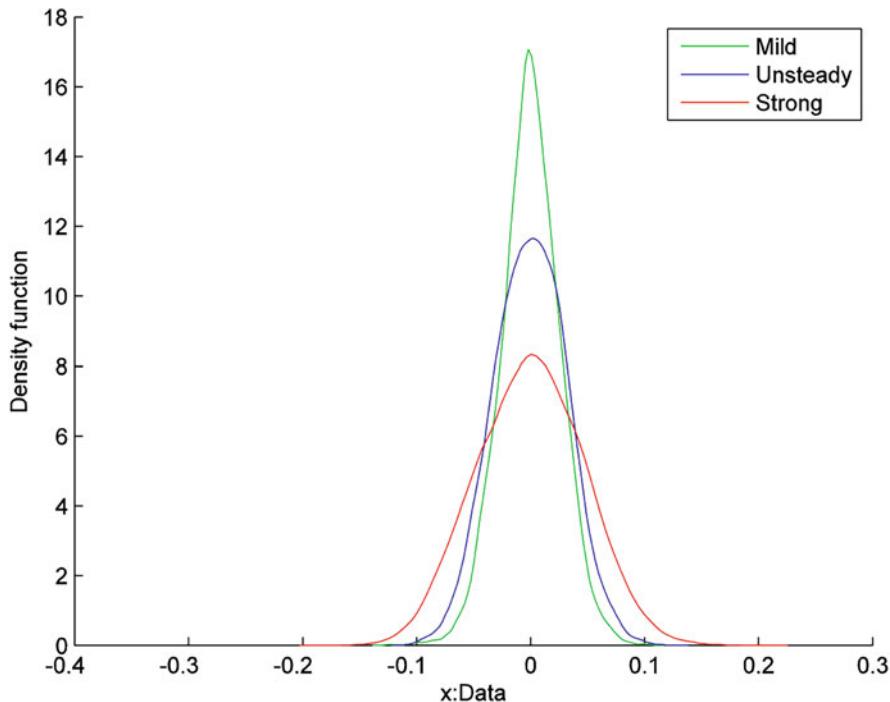


Fig. 16.10 Energy for the different noisy types

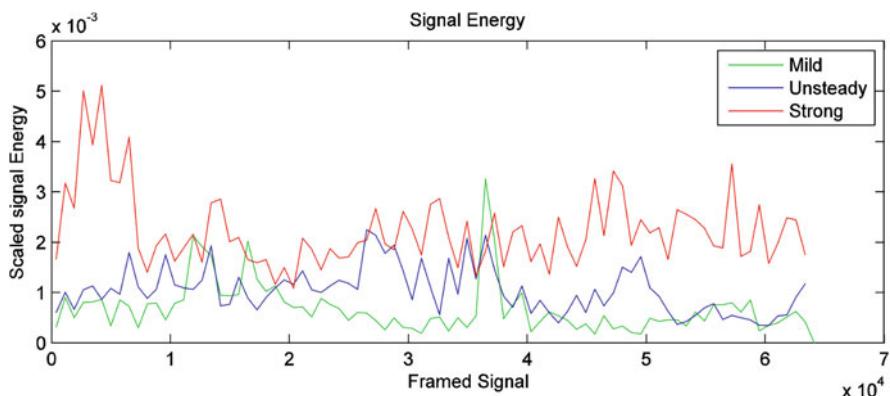


Fig. 16.11 Different noisy signal energies

16.9 Origin of Noises

One distinguishes between two kinds of origins with respect to our applications.

- internal origins:
 - Noise coming from speech production
 - Noise created from other parts of the body
- external origins:
 - Atmospheric noises
 - Noise coming from the earth from a seismic event
 - Industrial noises
 - Extraterrestrial noises

The noises by internal origin come from elements participating in the original sound production. The second kinds, the external origins noises, come from elements that are not part of the production. The specific form of each kind depends on the application. In the sequel chapters some application will be presented. Each presentation contains a part how to handle the noise.

16.10 Box Plot Evaluation

The box plot shows the distribution of the data in a descriptive way. That means they do not give us precise numbers but rather a qualitative overview. They are non-parametric and do not refer to any statistical distribution. However, they can partially visualize the noise. The box plot simply represents the data using its lowest value, highest value, median value and the size of the first and the third quartile. The median is counted by Eq. 16.8. There d_m is the depth i.e. the number of observations counted from the beginning of the ordered data set.

$$d_m = \frac{n+1}{2} \quad (16.8)$$

The depth of the first quartile is depicted in Eq. 16.9. There q_1 is the first quartile, i can be 2, 3 or 4, what is indicates in each corresponding quartile. It takes the value of the quartile depth.

$$d_{q_i} = \frac{in+2}{4} \quad (16.9)$$

The boxplot of the sound level depends on the measurement. Here we take A-weights. This is shown in Fig. 16.12. Next we come to another Boxplot example concerned with the levels of energy and sound and this is shown in Fig. 16.13. As it

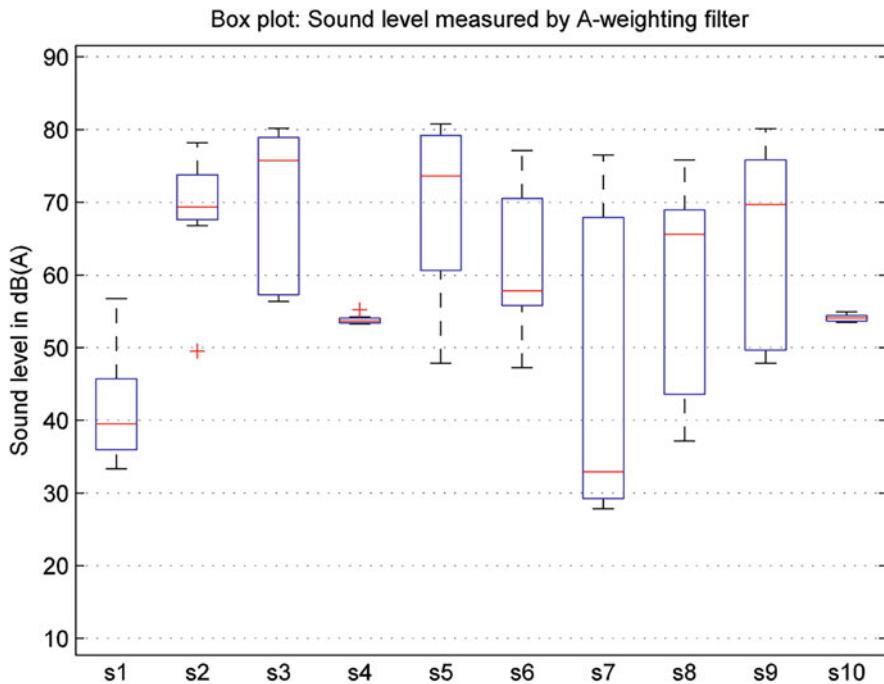


Fig. 16.12 Boxplot chart of sound level measurement by A-weighting filter

can be seen, the boxplot is useful for visualization as it displays noise level variation (boxed-in plot) as well is average value of the signal. Now we come to the individual types of noise.

16.11 Individual Noise Types

16.11.1 *Residual*

This type of noise is produced by machines. Typical example would be a classical telephone. It gives echoes that are not intended and disturb the recognition.

16.11.2 *Mild*

This type of noise is soft but always exists in the environment. It may make the communication annoying to the humans who cannot hear well or who may not have sharp hearing ability. Moreover, this soft mild noise is disturbing to some extent. The

noise is independent and modeled as white noise. Its mean it has zero mean expected value and unity variance, e.g., it is equal to one. All samples are assumed to be uncorrelated. Its power spectral density (PSD) is flat, and its first order probability density function (PDF) is Gaussian. It has the probability density function described by Eq. 16.10

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{\sqrt{2\sigma^2}}\right) = \frac{1}{2\pi} \exp\left(-\frac{x^2}{\sigma^2}\right) \quad (16.10)$$

This equation depicts normalized zero mean Gaussian process. The model is used for mild noise characterized by Gaussian process and is depicted by Eq. 16.11:

$$\sigma^2 = E\{|x^2[n]|\} \quad (16.11)$$

and it depicts variance of a mild noise.

16.11.3 Steady-Unsteady Time Varying Noise

This noise is continuous and ongoing in time. It can be steady or with a little variation in time. This is in contrast to the case of unsteady noise. Often, this noise comes from a running machine. It is characterized as a Gaussian process but its mean is not zero and the variance may not be always 1 as it is the case for the white Gaussian noise. If x is a Gaussian process for time instants n and $n = 0, 1, 2, \dots, N - 1$ and its mean is μ and its variance is one, then the pdf of the noise is characterized by Eq. 16.12 where $p(x)$ is the pdf of x .

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{\sqrt{2\sigma^2}}\right) \quad (16.12)$$

The noise model shown in Eq. 16.13 is an Auto-Regressive (AR) model but its parameters are obtained by the linear prediction by applying namely the Yule-Walker approach. In Eq. 16.13 the noise $d[n]$ is a linear combination of past i many β coefficients and a disturbance $w[n]$. This is assumed to be a white noise and it is weighted by g_b .

$$d[n] = \sum_{i=1}^q \beta_i d[n-i] + g_b w[n] \quad (16.13)$$

For treating this noise, one divides the signal first into sub bands using a cosine modulated quadrature mirror filter bank (QMF) and then the noise is minimized from each sub band by a spectral minimization technique. Afterwards the signal is enhanced in each band by Kalman filter. In this noise reduction, noise is varying in

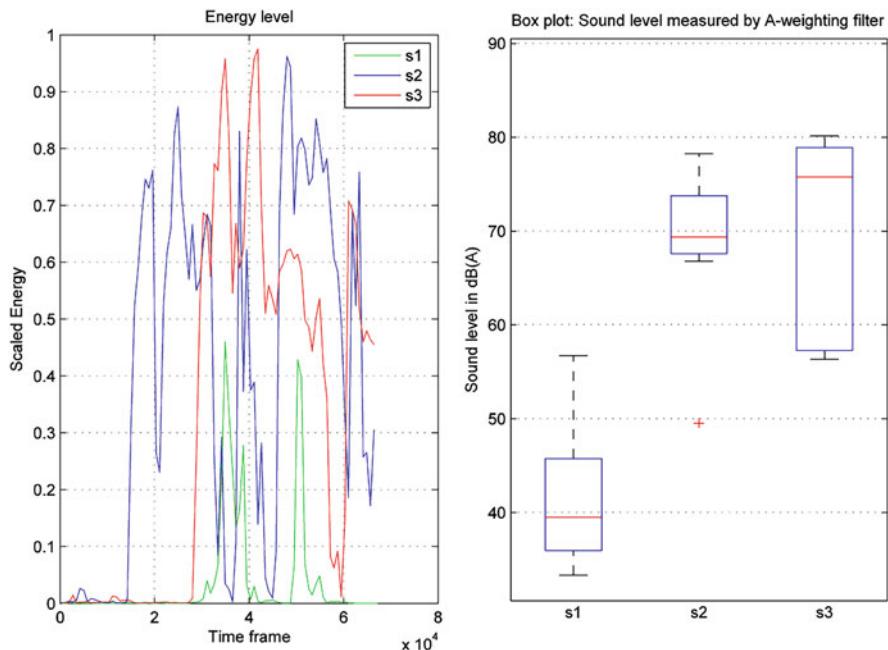


Fig. 16.13 Boxplot chart of sound level measurement by A-weighting filter

each sub band. In the Fig. 16.13 one sees the difference between an ordinary and an enhanced speech spectrum. In Fig. 16.14 we see different kinds of treatments. We show hybrid noise before and after treatment. Answer to the following general question is important: Why are we looking at these types? There is no precise and definite answer. Other types may occur and dominate a certain problem, what is unforeseen. However, in many if not most noise situations these types are the important noises. In addition, the methods for handling them are applicable for many other situations. Later on we will look at applications. Here just some short remark is given.

16.11.4 Strong Noise

This type of noise is extremely noisy such that humans have difficulties to understand each other. We do not deal with longer periods of strong noise because we assume that no signals arrive for that amount of time. Instead, we have two assumptions that allow something to be done:

- (i) The strong noise lasts only for a very short only period of time.
- (ii) The strong noise is randomly distributed and does not occur very often.

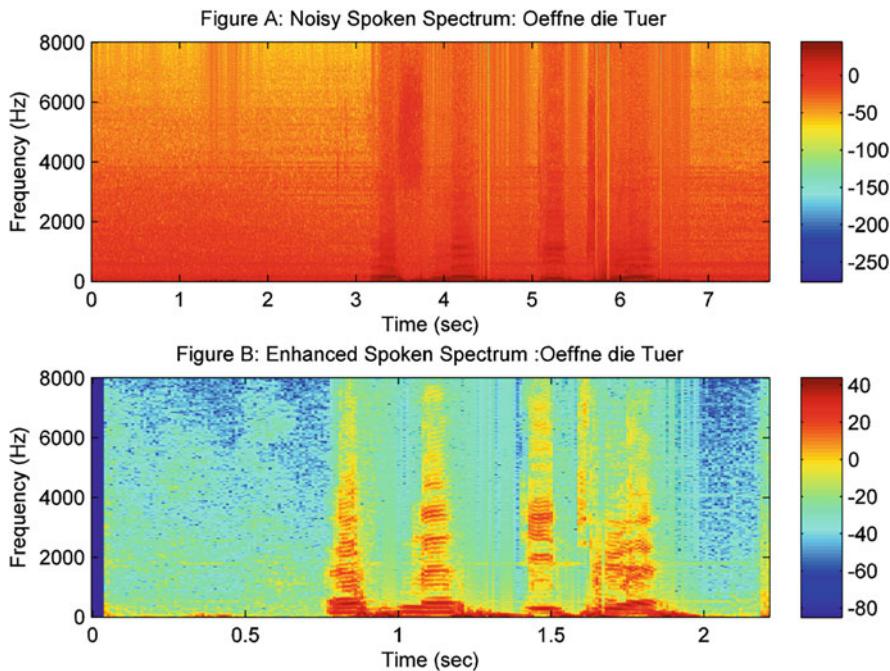


Fig. 16.14 Noisy and enhanced signal spectrum

These types of noise occur regardless of the types of the sources. Controlling the sources in general is not possible. By this, we mean we cannot keep production machines in the industry shut down in order to control the noise. Instead, we use a feedback as humans would do in such a situation. When noise from a loud airplane occurs during a telephone conversation a human typically asks the last sentence to be repeated. This will be shown below.

16.12 Solution to Strong Noise: Matched Filter

Because of large amplitude, short time duration, randomly and rarely occurring events, the strong noise is modeled as such—a shot. Their random characteristics are described by the Homogeneous Poisson distribution. We can control the noisy environment by applying speech recognition technology in the environment where humans can be replaced by machines. Treating the noise enhances the communication in the noisy environment, increases work place productivity and improves health care because people are not required to be exposed for very long period of time to negative effects of the noise. A machine can perform the task where the machine is part of the speech recognition system. We will describe events

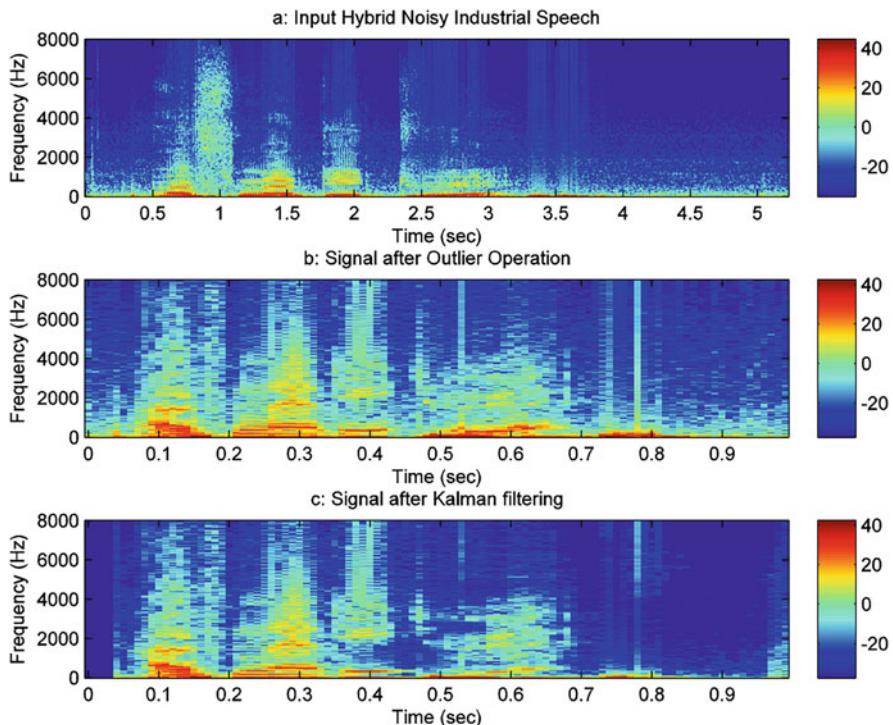


Fig. 16.15 Example of speech signal mixed with a strong noise and its handling

of strong noise as outliers. An outlier is an abnormal quantity in an observation that is often marked as a perturbation in the observation. Detection and removal of these outliers need a careful analysis so that the remedy of the outliers does not affect the signal. The remedy most often begins with the statistical information of the data such as the probability distribution of the data, a histogram or a boxplot of the data. A typical example of a strong noise is a sudden burst or a loud sound generated when a heavy object is falling down. In such case, we apply an approach which identifies the noise, detects it, and deletes it from the input for the machine recognition. This deletion may remove a part of the speech that needs to be understood. For circumstance, one solution is to provide a feedback to the “producer of the sound” (Fig. 16.15).

16.13 Background Information

General and Past

Noise was first cultivated in physics in the nineteenth century. Pehl (2001), gives a modern view on it. The first patent for a noise system was granted to Paul Lueg U.S. Patent 2,043,416 in 1934. There is much literature about the sources of

noise. Electromagnetic noise is discussed in Bell (1960). More on the formulations of the A-weighting filter in acoustic noise measurements, monitoring and control can be found in Bies and Hansen (2009). Another publication is Richard et al. (2004) and Wesfried and Wickerhauser (1993). For speech enhancement see Nguyen and Vaidanathan (1990). The de-emphasizing of the frequencies is discussed in Kolawole (2003). Details about box plot are in Tukey (1977).

Suggestion

Noise types have been considered with respect to their source, physical properties and colors. Electrical noise is found in Bell (1960). Noise reduction is discussed in Cohen et al. (2009). There are quite very many publications on the subject in this direction but not so much with respect to the impact of the receiver. A detailed information about A-Weights and their computation can be found in Lanman (2005).

16.14 Exercises

Exercise 1 Give Definitions of “loud” in terms of SNR.

Exercise 2 Give definition of a scenario where strong noise occurs. Show how you want to measure its distribution.

Exercise 3 Choose a specific Gaussian Process and compute $p(x)$ according to formula 16.12.

References

- [Bies2009] Bies, D. A., Hansen C. H. (2009). Engineering Noise Control: Theory and Practice. Taylor and Francis, KY, USA, 4th edition.
- [BELL1960] Bell D.A. (1960) Electrical Noise and Physical Mechanism. Van Nostrand, London.
- [Kolawole2003] Kolawole, M. O. (2003). Radar Systems, Peak Detection and Tracking. Newnes, Gordan Hill, GBR.
- [Richard2004] Richard L. St. Pierre, Jr., Daniel J. Maguire (July 2004).
- [Wesfried1993] Wesfried E., Wickerhauser M.V (1993). Adapted local trigonometric transforms and speech processing. 41. IEEE Transactions on Signal Processing.
- [Tukey1977] Tukey J.W. (1977). Exploratory Data Analysis. Addison-Wesley.
- [Pehl2001] Pehl E. (2001). Digitale und analoge Nachrichtenübertragung. Hüthig, Heidelberg.
- [Cohen2009] Cohen I., Huang Y., Chen J. (2009) Noise Reduction in Speech Processing. Springer.
- [Nguyen1990] Nguyen T.Q., Vaidanathan P.P. (1990). Structures for m-channel perfect reconstruction banks which yield linear-phase analysis filters. Acoustics, Speech and Signal Proc., IEEE Transactions, 1990.
- [Lanman2005] Lanman D.R. (2005). Design of a sound level meter. Lab. Report EN253 Matlab Ex.

Chapter 17

Reasoning Methods and Noise Removal



Overview

Noisy signals and their treatment have been studied for a long time. There have been many attempts for treatments but they have in common that they looked at each type individually. The goal was to remove the noise and enhance the original signals. One did this partially with a great success. The simplifying assumption was that only one type (mainly mild) of noise occurred. As a consequence many noisy situations could not be treated properly. For removing noise one needs reasoning methods. In this chapter we present such methods. We start with methods suitable for removing the mentioned types of noise. The removal of noise depends very much on the type of noise. A difficulty is that one cannot apply the removal methods in an arbitrary ordering. The application of one method may have a significant influence to other methods. At the end we present a general reasoning. This is Case-Based Reasoning that uses experience and similarity. It is in particular useful for stochastic processes.

17.1 Generalities

The term ‘Noise’ is subjective and what important about the signal process. Noise exists everywhere. Removal or reduction of noise is an important topic in all disciplines. Noise removal depends on the applications. For example, conversation or humans speech recorded automatically in a business or an office can be corrupted by the presence of noise. However, such noise may not significantly corrupt the speech or the desired signal. The term ‘desired’ is subjective and noise is very much related to the topic; namely what is desired and what is undesired.

A number of different typical noisy scenarios are considered here. Different type of noise handling methods are presented according to these noisy situations. These noise types have some commonalities. One cannot deal with the problem with general uniform techniques and we will handle them in a way which we call hybrid.

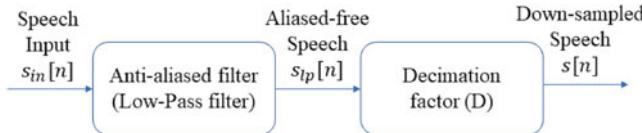


Fig. 17.1 Aliased-free down-sampler

Figure 17.1 shows a typical decimation process and how this applies to speech. This has actually been used in typical noisy situation as a pre-processing step and for the further processing. In Fig. 17.1 one sees that before down sampling two other operations take place. First a low-pass filter has been applied. This is an anti-aliasing filter. Then some decimation or down sampling or compression is applied.

17.2 Special Noise Removal Methods

In real world problem, some noises are mixed. In such cases, standard mathematical formulations may not be applicable where some hybrid methods can be preferable options. Some such options are described in this section.

17.2.1 Residual Noise

The removal of the residual noise is caused by a machine for instance a telephone and its removal is dealt with in the applied technique, that is microphones etc. We will not discuss this issue here.

17.2.2 Mild Noise

For this type of noise we use an Eigen analysis tool such as PCA (Principal Component Analysis) signal. The signal in this method is first decomposed into N number of frames. Then the auto-correlation of each frame is computed. The covariance matrix is the auto-correlation matrix when the signal is zero mean. The autocorrelation matrix is then decomposed into a matrix of eigen vectors and corresponding eigenvalues. One retains the significant components of the signal corresponding to the largest eigenvalues of the autocorrelation matrix whereas the noisy components corresponding to the smallest eigenvalues. Those smallest noisy component are discarded. The signal is then expressed as linear combination of its most significant principle components (largest valued components) represented

with Eigen vectors where the combination factors are normalized cross-correlation values of the signal and each PC Eigen vector.

Suppose we have $N \times M$ dimensional linear transformation matrix A whose elements are $a_{ir} = [a_{i1}, a_{i2}, \dots, a_{in}]$ in the N -dimensional i th column of A . We transform now the M -dimensional signal vector $s = [s_1, s_2, s_3, \dots, s_M]$ to an N -dimensional signal output $x = [x_1; x_2; x_3, \dots, x_N]$ through a linear transformation using matrix A , as depicted in equation below:

$$x = As \quad (17.1)$$

We assume that we have L samples of an N dimensional vector process $(s_0, s_1, \dots, s_{L-1})$. These vectors are our speech frames. Then PCA (see Sect. 17.6) is applied. The first step in PCA analysis is to obtain an estimate of the mean μ of the signal vector. This quantity is computed as usual by Eq. 17.2

$$\mu = \frac{1}{L} \sum_{m=0}^{L-1} s_m \quad (17.2)$$

An estimate of the covariance matrix C_{ss} of the signal vector is then obtained by Eq. 17.3 and the dimension of this matrix is $N \times N$.

$$C_{ss} = \frac{1}{L} \sum_{m=0}^{L-1} (s_m - \mu)(s_m - \mu)^T \quad (17.3)$$

The PCA is an Eigen analysis of the covariance matrix C_{ss} in terms of its eigenvectors matrix and eigenvalue vectors where the length of the Eigen vector is $N \times 1$. Hence we get:

$$C_{ss} = VAV^T = V^T AV \quad (17.4)$$

With letter s the vector process is indicated, with V an Eigen vector matrix with dimension $N \times N$, and with A the $N \times N$ diagonal Eigen value matrix are denoted. Since the covariance matrix is real and symmetric, its Eigen vectors are real and orthonormal and thus

$$x = Vs \quad (17.5)$$

Because the matrix V has the following property, the equation holds:

$$V^T V = VV^T = I \quad (17.6)$$

Since the covariance matrix is real and symmetric, its Eigen vectors are real and orthonormal. From this we obtain the equation:

$$C_{ss} = E(V_{ss}^T V^T) = V C_{ss} V^T = VV^T AVV^T = A \quad (17.7)$$

The transformation of s and the diagonalization and equalization of the covariance matrix C_{ss} resulted in Eq. 17.7.

$$x = \sqrt{C_{ss}} = (\sqrt{A}V)s \quad (17.8)$$

Finally, x is our desired enhanced signal process.

17.2.3 Steady-Unsteady Noise

The steady-unsteady time-varying noisy signal is a typical industrial noise. Often, this noise comes from a running machine. The noise level is categorized as being in between mild and strong. Conversations or activities can be affected in such a noisy environment. This noise occurs in a repetitive pattern and modelled by AR model in Sect. 17.1. The noise is characterized by Gaussian process $N(\mu, \sigma^2)$ with mean μ , and variance σ^2 and obtained its linear prediction parameters in Eq. 16.12. The linear prediction is introduced in part II of this book.

$$d[n] = \sum_{i=0}^q \beta_i d[n-i] + g_b w[n] \quad (17.9)$$

For treating this noise, the signal is first divided into sub bands using a cosine modulated quadrature mirror filter bank (QMF) and then the noise is minimized in each sub-band by a spectral minimization technique. This is a noise reduction in which the noise is varying in each sub band.

17.2.4 Strong Noise

The process representation will be somewhat different than the representation in general recognition, in particular with respect to the underlying probability. The reason is that we do not consider the production of signals in the form of digits only but we have to consider intervals too. This is due to the assumption that a strong noise occurs rarely and for a very short time only. We have modelled this in a probabilistic way in order to detect it. We will handle a strong noise event as a gunshot and model the probability distribution of the shots by a homogeneous Poisson model distribution described next.

17.3 Poisson Distribution

First we consider Poisson distributions of events in general. A stochastic process of events with a random variable \mathbf{X} is a Poisson process with a parameter $\lambda > 0$ and with functions indexed by k as the density functions:

$$f(k, \lambda) = \frac{\lambda^k}{k!} e^{-\lambda} \quad \text{for } k = 0, 1, 2, \dots \quad (17.10)$$

We assume for non-overlapping intervals (t_1, t_2) and (t_3, t_4) that the random variables $n(t_1, t_2)$ and $n(t_3, t_4)$ counting the occurrences of events in the intervals are independent. The parameter λ represents the average. If the expected number of occurrences of events in an interval is k , then the probability that there are exactly k occurrences is equal λ is given by Eq. 17.10. A Poisson process is a stochastic process governed by Poisson distributions.

17.3.1 Outliers and Shots

Relating to the idea of strong noise we first define the concept of an outlier, i.e. something that rarely occurs and for a short time only. For this we denote a noisy observation by $z[n]$. Then we can perform the following mathematical steps for handling outliers. Mathematical tasks:

- (i) Determine a threshold θ where $0 < \theta < 1$ defining the outlier interval.
- (ii) Compute the expectation $z_{ex} = E[z[n]]$
- (iii) Identify the set of “outlier” samples n by $|z[n] - z_{ex}| \geq \theta$ and $z[n] > z_{ex}$.
- (iv) Remove these identified “outliers” from the noisy signal.

This means that outliers differ significantly from the average. The identification of the outliers is an essential step to identify the intervals in which they occur. Removal means to remove these intervals from the signals. The identification of outliers is a standard but difficult problem in data analysis distribution values. This comparison is computationally very involved. Most methods analyzing training and test data and compare the difference in their probabilities. Many approaches make use of the fact that outliers are known. For the outlier identification we need to determine the elements of the mathematical definition from above. Here we have to take of the probability distribution of the shots which has not been defined yet.

17.3.2 Underlying Probability of Shots

Shots are introduced in different ways in digital analysis. A shot is an interval that satisfies the conditions (are small and randomly distributed) described for outliers. That means all samples in the shot exceed the threshold for the strong noise. It follows from our assumptions that the set of these samples is in fact an interval. An event will now be a shot and the process model describes the shots. In our modelling this process is a Poisson process, i.e. it has a Poisson distribution. The standard deviation of shot noise is equal to the square root of the average number of events N . Therefore the signal-to-noise ratio (SNR) is given by:

$$SNR = \frac{\sqrt{N}}{N} = \sqrt{N} \quad (17.11)$$

The signal-to-noise ratio is very large when N is very large. There are several examples also outside of speech. Prominent examples are:

- Tunnel junction that is characterized by low transmission in all transport channels, therefore the electron flow is governed by Poisson process.
- Quantum point contact that is characterized by an ideal transmission in all open channels, therefore it does not produce any noise.

17.4 Kalman Filter

We continue the discussion that we started in Chap. 12. One can estimate something before an event has happened (priori) or after an event has happened (posteriori). Using Kalman filters one can use observations (that can include errors) conclusions of the state. For this one has to make use of the mathematical structures of the basic dynamic systems. Formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The method works in two steps. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement is observed, these estimates are updated. Let's start with the Equation $y[n] = s[n] + b[n]$. It expresses the observation y in terms of the true value s and the disturbance b that are presently unknown. The state information is written in the state space form in Eq. 17.12 presented below. In the equation, n is the time varying index and the state vector s at time $n + 1$ is a p length vector of the linear combination of p previous vectors using a $p \times p$ dimensional matrix A and some additional disturbance vector u of length p that is modeled as zero mean random white noise which is perpetrated by a $p \times 1$ dimensional matrix g_s at time n .

$$s[n + 1] = A[n]s[n] + g_s[n]u[n] \quad (17.12)$$

and the following equation that describes the noise vector b at time $n = 1$ that can be considered linear combination of a $q \times q$ dimensional matrix $D[n]$ and noise vector b at time n with an additive disturbance v which is perpetrated by a $q \times 1$ dimensional matrix g_b at time n :

$$b[n + 1] = D[n]b[n] + g_b[n]v[n] \quad (17.13)$$

The vector v is modeled as zero mean random white noise. This can be expanded as:

$$\begin{aligned} s'[n - p] &= [s[n - 1] \cdots s[n - p]] \\ b'[n] &= [b[n - 1] \cdots b[n - q]] \\ g'_s &= [0, \dots, 1] \\ \gamma' &= [0, \dots, 1] \\ g'_b &= [0, \dots, 1] \\ \varphi' &= [0, \dots, 1] \end{aligned} \quad (17.14)$$

$$A[n] = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \alpha_2 & \cdots & \alpha_p \end{bmatrix} \quad (17.15)$$

$$D[n] = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1 & \beta_2 & \cdots & \beta_q \end{bmatrix} \quad (17.16)$$

The α_i and β_i are coefficients to be determined. How these coefficients are determined using the ULS approach and the Yule-Walker equation are explained in Sect. 13. For this an earlier figure is partially repeated. This shows disturbance matrix $G[n]$ times measurement disturbance $w[n]$. The noisy observation vector given is y at time $n + 1$. In Fig. 17.2, we see the signal flow diagram of the Kalman prediction and the estimation for the colored noisy signal. Here one sees that the observation consists of s and noise b . These are predicted first and estimated using the Kalman gain K in order to generate estimated s . In this diagram, we see the observation consists of mixed signals and noise. Both of them are modeled by AR approaches. The observation is estimated, updated and corrected by the Kalman gain matrix K . The input is hybrid noise. In the natural environments many uncorrelated noise types occur. The observation equation is given in Eq. 17.17 presented below:

$$y[n + 1] = C[n]y[n] + G[n]w[n] \quad (17.17)$$

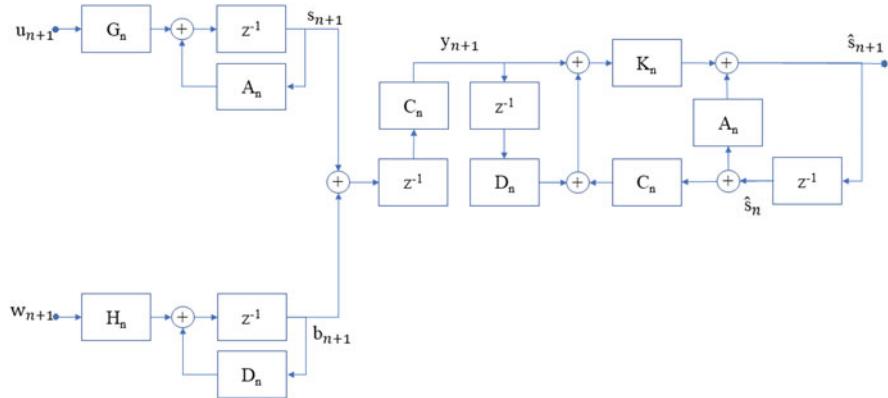


Fig. 17.2 Flow diagram of the Kalman prediction

In the Eq. 17.17 n is the varying time index. The equation says if we look at the observation vector y at time n , then we see that it is a linear combination using a $q \times p$ dimensional matrix $C[n]$ at time n from the previous q states of the state $s[n]$ and disturbance which is modeled as a colored noise b . In this equation, a measurement determined using the ULS approach and the Yule-Walker equation is explained in Sect. 12. The detailed version of Eq. 17.17 is:

$$C[n] = \begin{bmatrix} A[n] & 0 \\ 0 & D[n] \end{bmatrix}; G[n] = \begin{bmatrix} g_s & 0 \\ 0 & g_b \end{bmatrix}; y[n] = \begin{bmatrix} s[n] \\ b[n] \end{bmatrix}; b[n] = \begin{bmatrix} u[n] \\ v[n] \end{bmatrix} \quad (17.18)$$

There u_n and v_n are zero-mean white noise sequences. The covariance of u_n is $\sigma u[n]^2$ and the covariance of v_n is $\sigma_b[n]^2$ such that Eq. 17.17 holds.

$$\begin{bmatrix} \sigma u^2[n] & 0 \\ 0 & \sigma_b[n]^2 \end{bmatrix}$$

$W[n]$ is covariance matrix of $w[n]$. The state s and the noise b it is assumed to be uncorrelated. The noisy observation vector y at time n is a sum of signal s and a noise b . The state space definition of the observation y is rewritten to obtain the matrix $H[n]$. The observation vector the becomes as depicted in Eq. 17.19

$$y[n] = H^T[n] = \begin{bmatrix} \gamma[n] & \Psi[n] \end{bmatrix} \begin{bmatrix} s_n \\ b_n \end{bmatrix} \quad (17.19)$$

17.4.1 Prediction Estimates

The prediction estimate of $y[n]$ at time n given the value for $n - 1$ is $y[n|n - 1]$. $\hat{y}[n|n - i]$ is the predicted value of $y[n]$ based on the observation samples up to time $[n - i]$. In order to estimate the error we have to consider the development of the error over time. The innovation or the error signal $e[n|n]$ is provided in Eq. 17.20.

$$e[n|n] = y[n] - \hat{y}[n|n] \quad (17.20)$$

This in turn leads to Eq. 17.21 presented below

$$e[n|n - 1] = y[n] - \hat{y}[n|n - 1] \quad (17.21)$$

The prediction is shown in Eq. 17.22 presented next

$$\hat{y}[n|n - 1] = C[n - 1]\hat{y}[n - 1] \quad (17.22)$$

By using Eq. 17.22 the error can be derived:

$$e[n|n - 1] = y[n] - \hat{y}[n|n - 1] \quad (17.23)$$

17.4.2 White Noise Kalman Filtering

The Kalman filter is first applied to signals assuming the signals are corrupted by the white noise. The model is shown in Eq. 17.24 below.

$$\begin{aligned} x[n + 1] &= A[n]s[n] + B[n]w[n] \\ y[n] &= C[n]x[n] + v[n] \end{aligned} \quad (17.24)$$

The system noise $w[n]$ is a white Gaussian noise. This has zero mean and unit variance. The measurement noise $v[n]$ is an additive noise which is also zero mean and has known variance. Here the signal model is based on the Yule-Walker equation and the transition matrix A is a $p \times p$ dimensional coefficient matrix. The system matrix units B, C are $p \times 1$ and s is a vector ($p \times 1$ dimensions). In Fig. 17.3 we see that a time varying steady-unsteady noisy signal is enhanced. For this figure the $M = 32$ sub-bands where each band is of 1024 length was used. The noise is minimized and the Kalman filter is applied in each sub-band. In Fig. 17.3a is the noisy spoken German command “Offne die Tuer” (which is means “open the door”), and Fig. 17.3b is an enhanced version of this spoken command in the time domain is presented.

17.4.3 Application of Kalman Filter

This filter is used in practice, for instance in the acoustic control. Below figure present recording of a spoken command uttered in an noisy industrial environment. The command is collected at 48 kHz sampling rate spoken by a German native speaker in the hybrid industrial noisy environment. In Fig. 17.3a, the command is preprocessed with a standard first order low pass filter. In Fig. 17.3b, the noisy speech is preprocessed by special technique where the noisy command is first standard low pass filtered, then the redundancy is reduced and pre-emphasized and enhanced. In the pre-emphasizing, the speech is first decimated to 16 kHz as shown in the above block diagram. Then redundancy is removed by computing the envelop applying Hilbert transformation and selected a threshold below which the samples of the signal is removed, then the signal is smoothed by a low pass filter. The pre-emphasized signal is enhanced by decomposing the signals into sub-bands applying quadrature mirror filter-bank and then the Kalman filter for the colored noise. The Kalman filtering is applied for de-noising the speech signal in each sub-bands. After de-noising the decomposed signals are synthesized. This we see in Fig. 17.3b. The signal to noise ratio (SNR) in Fig. 17.3a is -7.689 and the SNR of Fig. 17.3b is 28.789 .

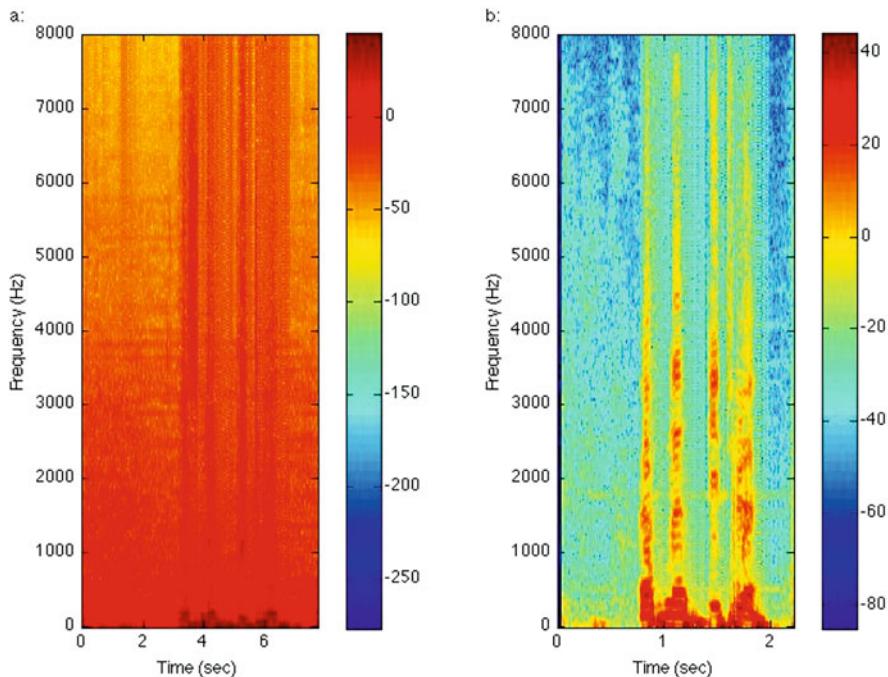


Fig. 17.3 Steady-unsteady noisy signal enhanced by Kalman filtering. (a) Noisy spoken spectrum: Oeffne die Tuer. (b) Enhanced spoken spectrum applying M-band Kalman filter Oeffne die Tuer

Colored Noise

Consider the linear stochastic system with the following state-space description:

$$\begin{aligned} x_{k+1} &= A_k x_k + \Gamma_k \xi_k \\ v_k &= C_k x_k + \eta_k \end{aligned} \quad (17.25)$$

where A_k , Γ_k , and C_k are known $n \times n$, $n \times p$ and $q \times n$ constant matrices, respectively, with $1 \leq p, q \leq n$. The problem is to give a linear unbiased minimum variance estimate of x_k with initial quantities $E(x_0)$ and $Var(x_0)$ under the assumption that:

- (i) $\xi_k = M_{k-1} \xi_{k-1} + \beta_k$
- (ii) $\eta_k = N_{k-1} \eta_{k+1} + \gamma_k$

where $\xi_{-1} = \eta_{-1} = 0$, $\{\beta_k\}$ and $\{\gamma_k\}$ are uncorrelated zero-mean Gaussian white noise sequence satisfying:

$$E(\beta_k \gamma_l^T) = 0, \quad E(\beta_k \beta_l^T) = Q_k \delta_{kl}, \quad E(\gamma_k \gamma_l^T) = R_k \delta_{kl},$$

and M_{k-1} and N_{k-1} are known $p \times p$ and $q \times q$ constant matrices.

The noise sequences ξ_k and η_k satisfying (i) and (ii) are called colored noise. One has to be concerned with converting the system of x to white. Therefore

$$z_k = \begin{bmatrix} x_k \\ \xi_k \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A_k & \Gamma_k \\ 0 & M_k \end{bmatrix}, \quad \hat{\beta} = \begin{bmatrix} 0 \\ \beta_k \end{bmatrix}$$

Which comes to:

$$Z_{k+1} = \tilde{A}_k z_k + b_{k+1} \quad (17.26)$$

17.5 Classification, Recognition and Learning

The recognition occurs quite often, but it is also quite difficult and by no means a routine task. The idea is to use learning for recognition. The situation for learning speech recognition is now as follows. Suppose one faces the conditions:

- There is a fixed list L of enumerated words.
- There is one or more agents who speak the words L many times. These are collected in a set S .
- There is an automaton that receives the spoken words.

The goal is that the automaton identifies the words from S correctly. This gives a classification on S : Two elements are in the same class if they correspond to the

same word in S . This splits S into a disjoint set covering the classes. Therefore one can rephrase the goal as: The automaton should discover the classes, i.e. solve the classification problem. This is asking for an unsupervised method of learning. One way to approach is to use clustering. The classes correspond to the ideal clusters. The goal for clustering is to come close to the ideal clusters. For a clustering method the learning properties depend on the specific method. In principle, every method can be taken as for instance K-means. Experience shows that one needs very many examples collected in S . This number increases drastically if there are several speakers or if some dialect spoken by the persons is present. On the other hand, the user has some influence too. This consists in the choice of the vocabulary. The user can choose the vocabulary to express different ideas. The words should be dissimilar enough so that no two words sound similar. This gives avoids many miss-classifications.

17.5.1 Summary of the Used Concepts

A summary of the used concepts is threefold:

Evaluation This computes the probability of the observations $p(o | \lambda)$ given the model λ and the probability of the observations of being in certain state at certain time by the forward algorithm. For this we employ different methods.

Search A preliminary remark is that search is a method that is underlying many machine learning and optimization procedures. Here search is used to compute the optimal likelihood of the observations and a state given the mode. It tries to find the best state holding the word on the sequence of features observations at a specific time the given model. Here we use the Viterbi search algorithm.

Learning and Re-estimation The re-estimation adjusts the model λ in order to maximize the probability $p(o | \lambda)$ of the feature vectors $o \in O$. This improves the initial HMM parameters estimation. These are done by expectation maximization algorithm.

17.6 Principle Component Analysis (PCA)

The Principal component analysis (PCA) is a procedure that covers a set of correlated observations into a set of values of linearly uncorrelated variables called principal components. It uses an orthogonal transformation. The denoising operation of our noisy speech is done by applying PCA. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as

possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The PCA is also a tool. In fact, it is an Eigen value analysis tool. This extracts the meaningful basis of the noisy redundant signal by searching for the principle components of the original signal and produces in a coordinate system using the coordinates of the transformed space in order to simplify a complex expression to a simpler one. The principal components (PC) are the linear combinations of the basis vectors. PCA is the simplest of the true Eigen vector based multivariate analysis. Often, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data. If a multivariate dataset is visualized as a set of coordinates in a high dimensional data space axis per variable), PCA can supply the user with a lower dimensional picture, a projection or “shadow” of this object when viewed from its most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced. The example of noisy signal before and after it is denoised is presented in Fig. 17.4. In the presented figure the German sentence, “Oeffne die Tuer” was used in Enhanced version as well as Special Enhanced PCA verin as in Fig. 18.11 was reused. On the left side there is a white noise input and on the right side the output after applying PCA. One should compare this figure with previously presented Fig. 17.3 above where steady-unsteady with Kalman filtering was compared. The difference is clear?

The option of applying this PCA denoised signal for the classification and recognition is investigated. The result of this PCA application is shown in Fig. 17.4. The first vector $\mathbf{w}_{(1)}$ has to satisfy:

$$\begin{aligned}\mathbf{w}_{(1)} &= \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (t_1)_{(i)}^2 \right\} \\ &= \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (x_{(i)} \times \mathbf{w})^2 \right\}\end{aligned}$$

Equivalently, writing this in matrix form it will lead to:

$$\begin{aligned}\mathbf{w}_{(1)} &= \arg \max_{\|\mathbf{w}\|=1} \left\{ \|\mathbf{Xw}\|^2 \right\} \\ &= \arg \max_{\|\mathbf{w}\|=1} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{w} \mathbf{X} \right\}\end{aligned}$$

Since $\mathbf{w}_{(1)}$ has been defined to be a unit vector, it equivalently also satisfies:

$$\mathbf{w}_{(1)} = \arg \max \left\{ \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{w} \mathbf{X}}{\mathbf{w}^T \mathbf{w}} \right\}$$

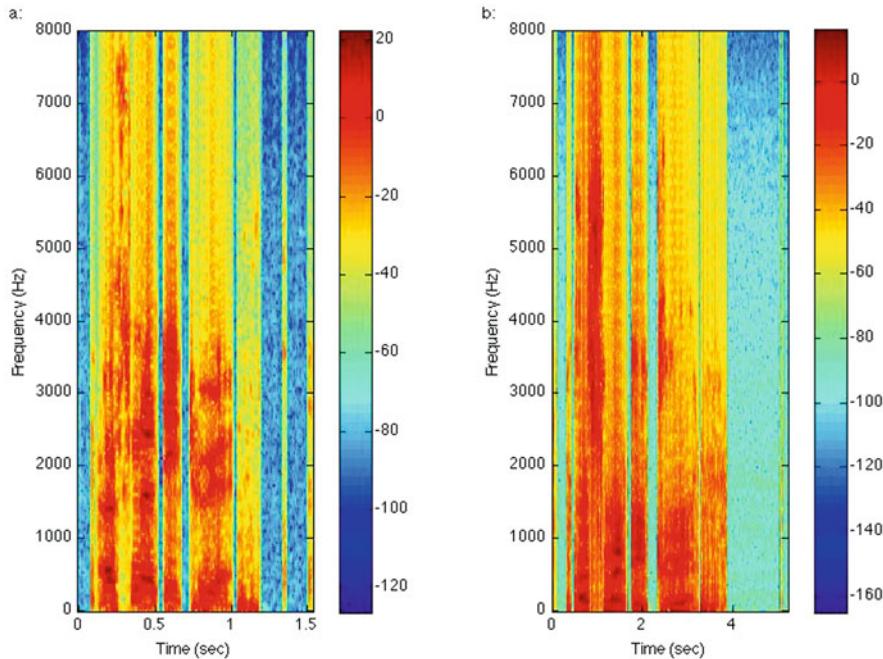


Fig. 17.4 Example of applied PCA to noisy signals. (a) Standard enhanced spectrum applying PCA: Oeffne die Tuer. (b) Special enhanced spectrum applying PCA: Oeffne die Tuer

A standard result for a symmetric matrix such as $\mathbf{X}^T \mathbf{X}$ is that the quotient's maximum possible value is the largest Eigen value of the matrix, which occurs when \mathbf{w} is the corresponding Eigen vector. With $\mathbf{w}_{(1)}$ found, the first component of a data vector $\mathbf{x}_{(i)}$ can then be given as a score $t_{1(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(1)}$ in the transformed co-ordinates, or as the corresponding vector in the original variables, $\{\mathbf{x}_{(i)} \cdot \mathbf{w}_{(1)}\} \mathbf{w}_{(1)}$. The further components are obtained in a similar way. A variant of principal components analysis is used in neuroscience to identify the specific properties of a stimulus that increase a neuron's probability of generating an action potential.

17.7 Reasoning Methods

Reasoning enables the user to discover information that is not directly visible. In this chapter the concern was on information disturbed by noise. Another goal is to find the information contained stochastic processes. The processes contain a message but one sees just the individual signals. The relevant information is contained in the whole process.

17.7.1 Case-Based Reasoning (CBR)

Case-Based Reasoning is an additional method for solving problems. It is based on experience. The experience is formulated in terms of solved problems that are called cases. The nature of the problems is that it does not need to have an optimal solution. This problem occurs frequently for instance in medical field. It is sufficient to know what “better” is. In those cases the tool of choice is the similarity measure. One may also know good and dangerous regions of the stochastic processes. Here one can employ the method of rough sets. In Case-based reasoning one considers a certain set Q of possible queries and a set S of possible solutions for the queries, A set $Cases \subseteq Q \times S$ such that for each query $q \in Q$ there is some unique $s \in S$ such that $(q, s) \in Cases$. On the solutions with respect to case, the user now has a quality measure. That means the user knows what a better solution is. The set Q will not contain each question q that could be formulated. The problem is to find for every question a solution from a set S . Here a new element of CBR comes in. It is the similarity measure sim comparing queries. This was discussed in Chap. 8 of Part II, in particular to the relation to utility. The procedure is: If a query q is given then proceed as follows:

- (i) Select q' from Q such that $sim(q, q')$ is maximal
- (ii) Take (q', s_{fromS}) from the set of $\in Cases$ as a solution after a possible modification.

There is now the last element of CBR for getting a solution that will be presented. Suppose now that one has a stochastic process P . As an example we take a medical process. These processes are discussed in the next chapter.

17.8 Background Information

General and Past

There is a variety of reasoning methods which handle noise. The concept of shot noise was first introduced in 1918 by Walter Schottky who studied fluctuations of current in vacuum tubes. See Schottky (1918). In the early studies, impulsive noise, shot noise or higher noisy situations have been considered in many areas. Some of them are acoustic studies, image processing, electrical and information theory. Look at Lowen (1990), Blanter and Büttiker (2000), Blanter and Büttiker (2000), Neenu Raju, Karunakaran (2013), Neenu Raju, Karunakaran (2013). In some studies median filtering is used to control such noise, see Heng Liu (2012). In other studies, some events are handled and controlled using a Stochastic-Poisson process. In some communication studies, the impulsive noise is considered as shot noise. See also Lim and Oppenheim (1979) and Lowen (1990).

Suggested

For applications as in steady-unsteady noise have a look at Lim and Oppenheim (1979). For noisy speech enhancement see Mustiere et al. (2008). For PCA we recommend Vaseghi (2008). There the goal is to improve robust feature extraction. The details of Principle Components are also described in Tsukuba (1993). The Kalman filter is named for honoring Rudolf E. Kalman, one of the primary developers of its theory; see Kalman (1960). The Kalman technique became popular in particular in control theory. It was used for situations where there is some imprecise measurement and therefore the observations have to be improved. More information on Kalman filtering is in Gelb (2001) and Paliwal and Basu (1987). The PCA approach was used in Takiguchi and Ariki (2007). See also Vetter et al. (1999). The case-Based Reasoning is a wide spread method including many industrial applications. A common tool is “CBR Works”. A complete description of CBR including theory and applications is in Richter and Weber (2013). For similarity see Part II.

17.9 Exercises

Exercise 1 Suppose the strong noise events come from closing a door in an office that is quite noisy. Model this as shots with a Poisson distribution.

Exercise 2 Describe precisely a situation with steady-unsteady noise.

Exercise 3 Describe in detail a situation with strong noise where no speech is involved.

Exercise 4 Write an algorithm that selects q' from a set Q such that $sim(q, q')$ is maximal for $q \in Q$.

References

- [Schottky1918] Schottky, W. (1918). “Über spontane Stromschwankungen in verschiedenen Elektrizitätsleitern”. Annalen der Physik (in German) 57.
- [Blanter2000] Blanter, Ya. M., Büttiker, M. (2000). “Shot noise in mesoscopic conductors”. Physics Reports (Dordrecht: Elsevier) 336.
- [Lowen1990] Lowen, Steven B. (1990): Power-law shot noise, ser. 6, vol. 36. Information Theory, IEEE, 1990.
- [Lim1979] Lim J.S., Oppenheim, A.V. (1979) Enhancement and Bandwidth Compression of Noisy Speech. Proceedings of the IEEE Vol. 67.
- [Mustiere2008] Mustiere, F., Bolic, M., Bouchard M. (2008) Improved colored noise handling in Kalman-based speech enhancement algorithms. IEEE, CCECE 00500, Ontario, Canada.
- [HengLiu2012] N.Z. Heng Liu (2012): An improved filtering algorithm based on median filtering algorithm and medium filtering algorithm, IEEE, 2012.

- [NeenuRaju2013]** Neenu Raju, Karunakaran, V (2013): An efficient method for removing impulse noise from the image, International Journal of Computer Science and Management Research, no. 3, vol. 2, Marc.
- [Paliwal1987]** Paliwal K.K., Basu A. (1987). A speech enhancement method based on Kalman filtering. 12, 180. IEEE, ICASSP, 1987.
- [Tsukuba1993]** Tsukuba (1993). Signal Modelling Techniques in Speech Recognition. Research and Development Center IEEE. 2013.
- [Vaseghi2008]** Vaseghi, Saeed V (2008). Multimedia signal processing: Theory and applications in Speech, Music, and Communications, Wiley, USA.
- [Kalman1960]** Kalman, R. E. (1960): A New Approach to Linear Filtering and Prediction Problems. Transaction of the ASME, Journal of Basic Engineering.
- [Gelb2001]** Gelb, A. (editor) (2001). Analytical Sciences Corp-Technical. Applied Optimal Estimation. 16. Edition. M.I.T. Press, Cambridge.
- [Takiguchi2007]** Takiguchi T, Ariki Y, (2007) PCA-Based Speech Enhancement for Distorted Speech Recognition. Journal of multimedia, Vol. 2, no. 5.
- [Vetter1999]** Vetter R., Virag N., Renevey P. Vesin J.-M. (1999) Single Channel Speech Enhancement Using Principal Component Analysis and MDL Subspace Selection. Eurospeech, 1999.
- [Richter2013]** Richter, M.M., Weber, R. (2013): Case-Based Reasoning. A text book. Springer Verlag.

Chapter 18

Audio Signals and Speech Recognition



Overview

We have referred to speech several times and will now provide a rigorous treatment of this subject. Audio signals are intended to be received by the humans. This view is also presented in Chap. 20 where we discuss the human ear. The major two types of audio signals are speech and music which are discussed in this chapter. We are building on the material where a central point is the study of transition probabilities of stochastic processes of signals.

18.1 Generalities of Speech

Although when speech is recorded, it is represented by the set of number representations (see Chap. 1). However, it would be too simple to view it as an ordinary sequence of digits. Two major reasons for this are:

- (i) The signals are of different intensity/loudness.
- (ii) There are subtle differences between the signals that are relevant.

The main types of variations in speech stochastic processes are the variations in the spectral-frequency composition and the variations in the time-scale. In the state transition probabilities, one can infer something about the transitions among the states and the variations of the duration on time-scales of the signal in each state. For example, the short or slow articulation can be expressed by self-loop transitions in the states of the model where the fast speaking or articulations can be skipped in next state connection. The state observation probabilities models the probability distributions of the spectral composition of the signal segments that are associated with each state. The next two illustrations show the hearing process by a human and when a machine is involved in the recognition. These are submitted to the listener who receives them. The ear of the listener performs various transformations.

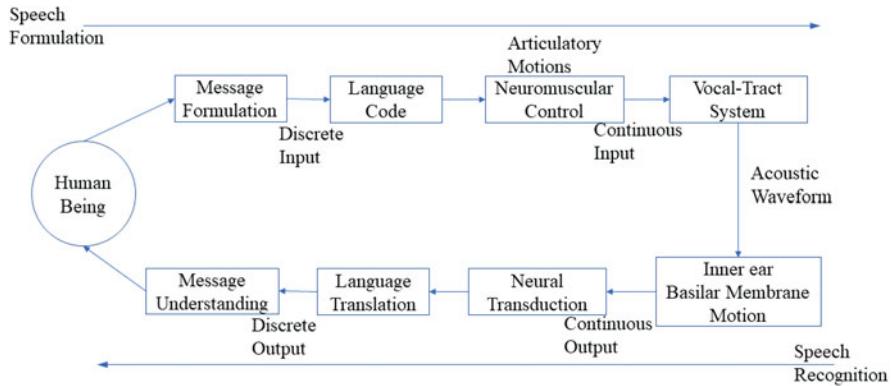


Fig. 18.1 The speech cycle from human perspective

The result is given to the brain for proper understanding. Today, machines can support and even replace many functions that were originally and exclusively used by humans. The upper line in the presented Fig. 18.1 describes speech generation, and the lower line describes speech recognition. As it is seen, the transfer from generation to reception is done in the form of waves. They have to be transformed into **signals** for further treatment.

The speech production process begins when the speaker formulates a message that is in the speaker's mind. It contains what the speaker wants to say to the listener. The first step in the speech generation level is discrete so we can readily estimate the rate of information flow using Shannon's information theory. A message can be represented as a sequence of discrete symbols. They are quantized into bits and the rate of the transmission of the information is measured in bits/second (bps).

The next step is the language code. This converts the message into, for example, some text that may represent phonetic symbols. The speaking tone of the person also plays a role. The latter includes the stress and duration information to describe the basic sound in the spoken format in a manner so that the speed and the emphasis of the message in the speech sound are well coded by the corresponding language. The result of the message formulation or the conversion is then sent to the neuromuscular controls. Next the neuro-muscular movement takes place to control the vocal apparatus for example the vocal folds, and/or the nasal part, or the lips that are needed to be moved to generate the message.

The outcome of the whole process is a continuous time analog waveform at the lips, jaw, velum etc. Thus the speech waveform is produced. The phonemes may be realized as a fundamental unit of speech sound. The pitch can be naively realized as the fundamental frequency of the speech sound. The vibration rates of the vocal folds during the speech production while transmitting sound through the vocal-tract are different. The message interpretation shown at the bottom left corner in Fig. 18.1 depicts process of speech perception and the speech recognition. The features are decoded and processed by the computer and/or human brain.

18.2 Categories of Speech Recognition

There are the following categories to be considered in speech recognition:

- **Isolated Speech Recognition**—**Connected Word Speech Recognition**
- Read-speech recognition—Spontaneous-speech recognition
- Speaker-dependent recognition—Speaker-independent recognition
- Closed-vocabulary recognition—Open-vocabulary recognition
- Close-talk recognition—Distant-talk recognition

Figure 18.2 depicts a machine supported speech recognition. The machine accepts the signal and performs the transformations that originally have been done by the human ear. This recognition is the part which we want to automate. The first step here is an effective conversion of the acoustic waveform into its spectral representation. For humans this takes place in the inner ear by the basilar membrane. Figure 18.3 shows a general structure of a speech recognition system. The figure depicts the scoring as applied in general speech recognition application. Two components stand out: “Acoustic Model” and “Lexicon” on one end, and on the other the “Language Model”. The Lexicon is necessary to define the number of words that can be recognized—similar to dictionary. An example is Digits and Numbers, or, as in general speech recognition, any number of the words that the recognition is capable of recognizing. In addition to the words the Lexicon specifies corresponding possible pronunciations of that word. Acoustic Model requires significant number of real data collected from actual examples of the speech to be presented to the system for training purposes. Each acoustic model should provide the link between to orthographic pronunciation of a particular word presented in the wavfile. Finally the language model provides prediction probability of next words, as for example in the sentence “I want to make a phone ?”. Clearly the next word

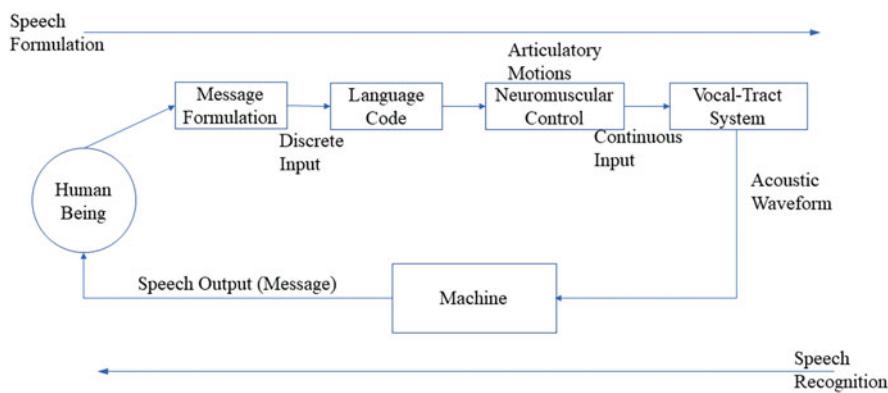


Fig. 18.2 Speech recognition by a machine

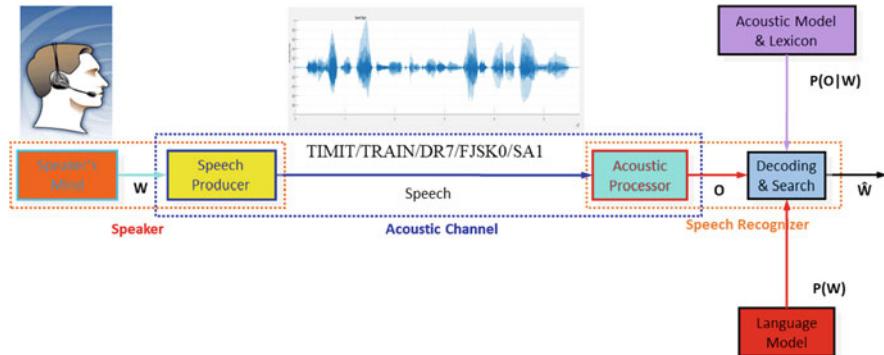


Fig. 18.3 General speech recognition system

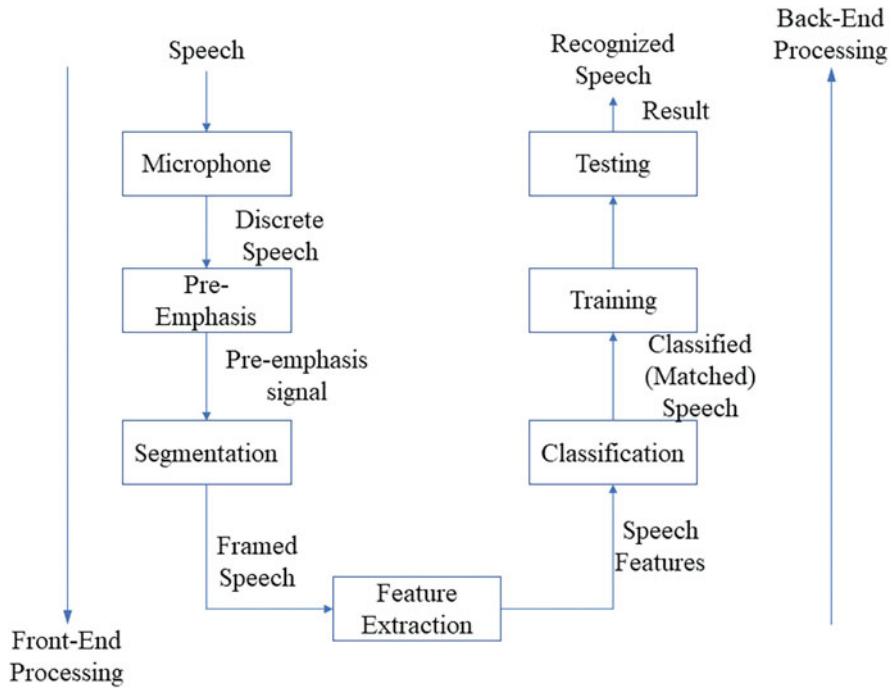


Fig. 18.4 A more detailed speech recognition system

is “call” and it is not “cake”. In the sequel we will discuss the details of training of acoustic models as exemplified in Fig. 18.4 presented below.

We consider the speech as a stochastic process and make use of the concepts and techniques developed in this area. The HSM with respect to the speech recognition problem is described by Fig. 18.5. This figure shows how the HSM fits to the Bayes’ rule in order to solve the speech recognition problem.

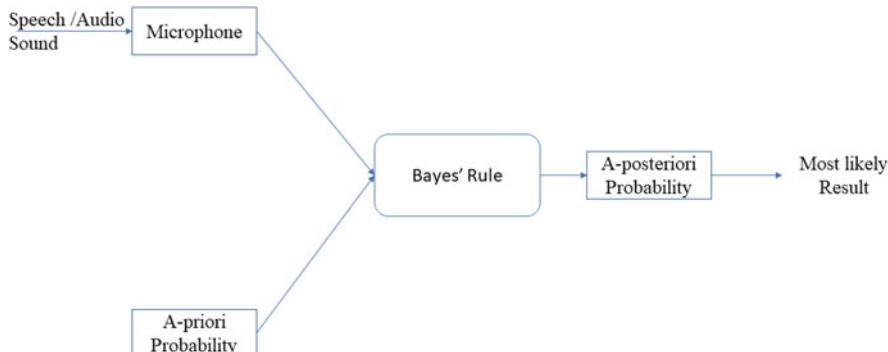


Fig. 18.5 Bayes rule in speech recognition system problem

A final question is when the Bayes' theorem should be applied? Part of the challenge in applying Bayes' theorem involves recognizing the types of problems that warrant its use. One should consider Bayes' theorem when the following properties of a problem are present.

- The sample space is partitioned into a set of mutually exclusive events $\{A_1, \dots, A_n\}$.
- Within the sample space, there exists an event B , for which $P(B) \geq 0$. The task is to compute a conditional probability of the form: $P(A_k|B)$.
- At least one of the two sets of probabilities described below are known.
 - $P(A_k \cap B) \forall A_k$
 - $P(A_k) \& P(B|A_k) \forall A_k$

18.3 Automatic Speech Recognition

The speech recognition can be of different types and they all are kind of complex. Thus the architecture and structure of the Automatic Speech Recognition (ASR) can be varied. On a superficial level the difficulties are not visible. Here we provide some examples of possible [Automatic Speech Recognition](#) types and their architecture.

Influencing Parameters of Automatic Speech Recognition: Speech recognition factors and types as well as background information about speech production and the role of the human ear in speech perception for representing speech for a speech recognition system are central.

18.3.1 System Structure

The acoustics is a major focus of the structure in sound propagation and thus speech recognition. Understanding the acoustics is a first necessary step but it says nothing about the meaning. We consider one word, or several words, sub words or several sub words as a unit where linguistic details are not a major focus. This searches the information content of the word present in a waveform. The waveform determines the acoustics because the waves are the elements that reach the ear for further processing. This requires a bit of understanding of biological structure that is utilized in the process. Based on understanding how humans ear functions the present machine structures have been built. We discuss this more below. The structure of the [Automatic Speech Recognition](#) can be:

- Continuous: Speech that is naturally spoken.
- Discrete: Here the speaker uses just one word at a time.
- Isolated: Only single words are used.
- For each structure one needs corresponding appropriate methods.

The type of an ASR can be

- Speaker dependent: A speaker dependent system is intended for use by a single speaker. The system depends on the speaker.
- Speaker independent: A speaker independent system is intended for use by many (possibly any) speaker. It is more difficult than one for use by a single speaker because it has to consider more variations. It may involve a collection of many thousands of sample data.

The vocabulary of an [Automatic Speech Recognition](#) can be:

- Small vocabulary: Order of tens of words
- Medium vocabulary: Hundreds of words
- Large vocabulary: Thousands of words

The vocabulary size is not discussed here, hence, in the applications considered it will play no role. For the knowledge about a language one distinguishes between productive and receptive. The former ones refer to speaking and the latter ones to reception. Another related distinction is between oral and/or written vocabulary.

18.4 Speech Production Model

Here we complement the issue of the human speech production and its use for computations. The purpose of the computational speech production model (also called source excitation model) is to manipulate the reality computationally and to estimate also the constraints and the constants involved. This correlates the physical process to a computational model for the processing. The constraints in this context

are the natural regulations in generating the human speech and the constants are the weights, or the gains, or the speech parameters and the outputs that it generates through each pass and processing. This makes the ultimate speech unique for an individual person. The final formulation of the speech production model is the vocal-tract model. In the vocal-tract system the speech sound is produced through the opening and closing of the vocal folds. This introduces a vibration in the system, e.g., source of the sound.

18.5 Acoustics

The acoustic phonetics studies the acoustic properties of the speech and how these are related to the human speech production. A standard computational speech production model is discussed in later section and makes use of the study of the acoustics, phonetics, psychoacoustics and digital signal processing. The purpose of the computational speech production model is to manipulate the reality computationally and to estimate the constraints and the constants that describe the production in the human body. This correlates the physical process to a computational model for the processing. The constraints in this context are imposed by laws of nature. They are natural rules that limit, for example, speed of generation of the human speech. They exhibited as constants and weights on the speech parameters. Next we present computational aspects about some basic components used in the model in an overview. They are concerned with both, the human body and the machine. The vocal-tract is playing a vital role in the speech production. In the next description we partially use some qualitative terms to quantify the production.

18.6 Human Speech Production

Speech was created since the inception of the human race! In contrast—writing is at most a few thousand years old. It is available to anyone and everyone who learns to speak without formal instruction and attains a comparable level of skill and fluency. Also, speech is the most common and most natural manifestation of a language. Phonetics, is the study of speech sounds, and it is the bedrock of the scientific study of language. Henderson in 1877 said that “The form of language is its sounds”. Hence speech is expression of a language. ‘Language is not a cultural artifact that we learn the way we learn to tell time or how the federal government works. Instead, it is a distinct piece of biological makeup of our brain. Steven Pinker “The Language Instinct: How the Mind Creates Language”’.

Language, an expression of speech, is a complex, specialized skill, which develops in the child spontaneously, without conscious effort or formal instruction, is deployed without awareness of its underlying logic, is qualitatively the same in every individual, and is distinct from more general abilities to process information

or behave intelligently. For these reasons some cognitive scientist have described language as a psychological faculty, a mental organ, a neural system, and a computational module. The term “instinct” is preferred to all the above.

We study the human speech (hence human language) and how it is produced because we can model it in a way engineers do modeling. It will be complemented further when we discuss the human recognition system. The human production system allows to representing the speech production formally. After that we will use this for solving the recognition problem which is our main task. The real production of speech is organized by the human brain (which we will not consider here). There are several elements that participate in the production itself. These are of interest when one is constructing a speech system. First we provide a picture showing the major participants of the speech production in the human body. The elements shown in Fig. 18.6 are used for understanding how speech is produced. This is an engineering task and therefore these elements have to be described in an engineering style. This involves, the building of the model of sound production that can be analysed by using the mathematics of fluid flow mechanics, and acoustics. However, for our purposes, a useful approximation is to substitute a detailed model with a simplified one so called source/system model presented next.

18.6.1 *The Human Speech Generation*

In speech generation several parts of the human body participate, Fig. 18.6. Each part plays a special role in the generation of speech. The type of speech is reflected in the type of excitation that produces speech. In order to understand speech production some understanding of acoustic properties of sound are in order. The acoustics properties of the sound, are the topic of study of the speech and how they are related to the human speech production system. The most important acoustic properties of the vocal-tract is the resonances of the vocal-tract. That is, the sound originates in a similar manner as it originates in wind instruments. The modeling of such an acoustic wave is done through resonating standing waves in an air tube. The excitation of the tubes is done based on the sound that is being produced: voiced (e.g, glottis) or unvoiced (e.g., friction of air along the vocal tract tube) or combination of both. The sound is expelled from the lungs and is denoted as $u_s[n]$. This is considered the source of the speech. This source is then modulated, transformed through the larynx, the vocal fold muscle movements, then through the vocal-tract and finally through the lip and/or nasal cavity. These include the source excitation, the vocal-tract shaping, and the effect of the speech radiation at the lips.

The vocal-tract vibrates during the voice sound while it does not vibrate for some fricatives. The vibration is assumed to be quasi-periodic, while frication is model with noise type of sounds. The above mentioned excitation’s types are simplified to the voiced and the unvoiced types and they are used in a system by a simple efficient computational model to reflect the speech production process.

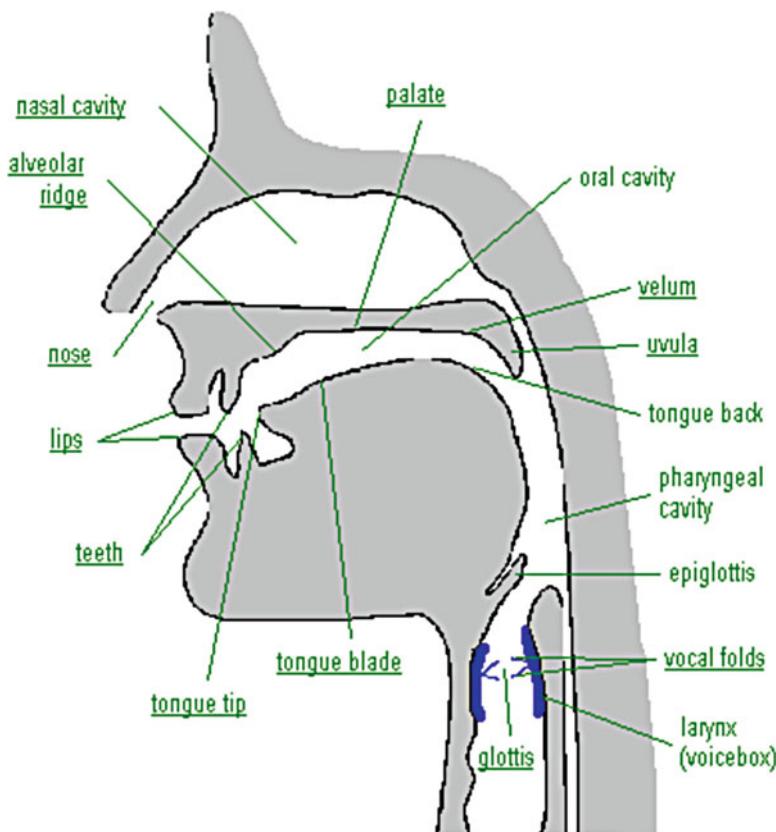


Fig. 18.6 The human parts involved in speech production

18.6.2 Excitation

One distinguishes three different kinds of excitation. The three excitation types are:

- **Periodic:** This produces the sound of *voiced speech*. Vocal tract is exited by periodic puffs of air by glottal vibration (e.g., opening and closing phases of glottis).
- **Random Noise:** This generates the *unvoiced speech*. This type of the sound is produced by constriction of vocal tract that in turn creates turbulent flow of air that produces unvoiced speech created by friction of the sound.
- **Combined Periodic with Random Noise:** Examples of this type of the sound are termed voiced-fricatives.

From the produced speech sounds point of view, the following categories can be used as further breakdown of the produced sound:

- **Voiced.** Example: The letter /l/ sound in the utterance of "six",
- **Unvoiced.** Example: The letter /s/ sound in "six",
- **Mixed.** Example: The sound corresponding to the letter "z" in the phrase "three Zebras" (e.g., "z" is voiced-fricative).
- **Impulsive:** This is type of signal is attached to the *plosive speech*. A plosive example is the sound corresponding to /t/ as in "pat", the /b/ as in "boot", etc. The list of plosive sounds of English, voiced and unvoiced, are provided next: b, d, g for voiced and p, t, k for unvoiced counterparts.
- **Whisper** is the sound uttered with glottis severely constricted and vocal folds do not vibrate. The sound is produced with turbulent flow of air.

There are over 40 speech sounds (e.g., Phonemes) in American English which can be organized by their basic manner of production (Table 18.1).

The classification of speech sounds (Phonemes of American English) is shown in the diagram presented in Fig. 18.7 below:

The source of the voiced speech is quasi-periodic as depicted in Fig. 18.9. The quasi-periodicity shows a repetitive tendency over a short time interval but it is not exactly the same as a periodic signal. The excitation source $u_s[n]$ follows through the glottal flow generator $f_g[n]$ (or the glottal filter) and it is modified by the gain factor g_v . This is how we model the glottal pulses exciting the vocal tract. For our

Table 18.1 The list of American English Phonemes

Manner class	Number
Vowel	18
Fricatives	8
Stops/plosives	6
Nasals	3
Semivowels	4
Affricates	2
Aspirant	1
Total	42

Phonemes of US English

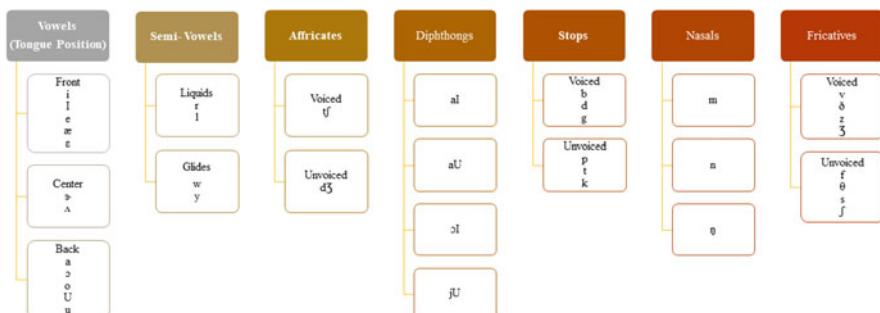


Fig. 18.7 Basic US English sounds classification

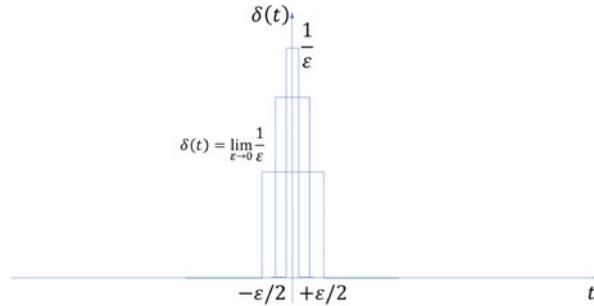


Fig. 18.8 Delta functions

purposes we have to go deeper into the structure of speech and how it is produced by humans. The input (e.g., excitation) for Voiced type of sound (e.g., vocalic) is quasi periodic train of pulses denoted here as $g_g[n]$. This train of pulses then passes through vocal tract that is based of its configuration and a specific sound is being produced. This process is modeled by a vocal-tract filter. When the signal is passed through the vocal-tract filter, denoted here as $f_v[n]$, it generates the sound denoted as $v_f[n]$. Sound is produced by passing through the lips of the speaker. This is modelled by radiation filter $f_r[n]$ that we perceive as the voiced speech $s[n]$. The process is shown in Eq. 18.1. That equation depicts train of pulses repeated every P samples which models voiced speech.

$$u[n] = \sum_i \delta[n - iP] \quad (18.1)$$

The δ function is used to model train of pulses. Since δ function is theoretical abstraction, and hence this function is approximated by the following signals as depicted in the next Fig. 18.8:

The formulation of the periodic pulse source and its voiced speech production is presented in Figs. 18.9 and 18.10. In Fig. 18.9 each vertical arrow headed line at the left corner indicates a train of pulses which are later weighted by the gain. It is assumed that the pulses are periodic.

18.6.3 Voiced Speech

The voiced speech is generated when the vocal-tract is excited by a series of periodic pulses. The variation in the voiced speech is very smooth within a period. For this reason, it is analyzed as essentially a periodic signal. If the input speech looks visually nearly periodic then it is termed as voiced speech. In the production of this speech, the air coming out of lungs through the trachea is interrupted periodically by the vibrating vocal folds. By this process, the glottal wave is generated that

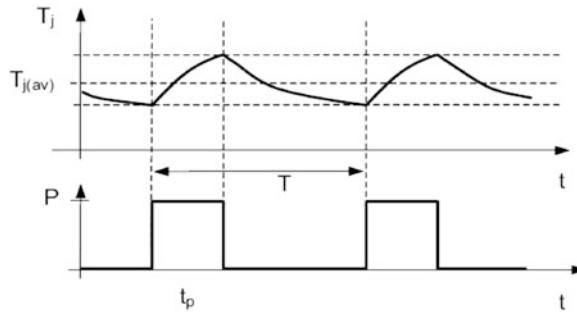


Fig. 18.9 Periodic pulses

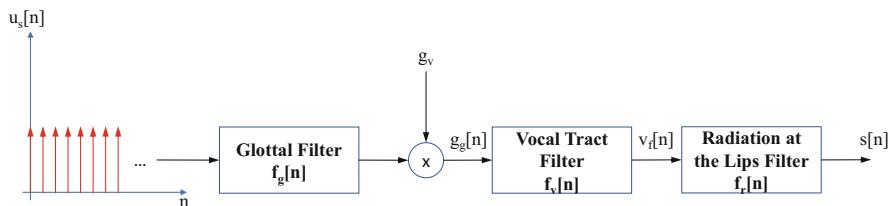


Fig. 18.10 Source model of periodic pulses



Fig. 18.11 Voiced speech in the source excitation model

influences the speech production system. The speech is modulated by passing through vocal tract that results in the voiced speech. Figure 18.10 visualizes the vocal speech production (Fig. 18.11).

The vocal tract filter for the voiced source given in Eq. 18.3 is a multiplication of the gain factor g_v and the transfer functions of the glottal filter $F_g(z)$, the vocal-tract filter $F_v(z)$, and the lip radiation filter $F_r(z)$ depicted below.

$$H(z) = \frac{S(z)}{U(z)} = g_v F_g(z) F_v(z) F_r(z) \quad (18.2)$$

Modeling of the sound uses a simplified all-pole filter. The transfer function of poles a_i . Thus the simplified equation of 18.2 is provided in 18.3,

$$H(z) = \frac{g_v}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (18.3)$$

indicating that the model of the transfer function of the vocal tract has p poles.

The input $u_s[n]$ of n is on the x -axis with an amplitude on the y -axis. The corresponding equations is 18.4 where the sign “ \otimes ” is used as the symbol for the convolution operation:

$$\begin{aligned} g_g[n] &= g_v(f_g \otimes u_s)[n] \\ v_f[n] &= (f_v \otimes g_g)[n] \\ s[n] &= (f_r \otimes v_f)[n] \end{aligned} \quad (18.4)$$

where $u_s[n]$ indicates vocal fold vibrations (e.g., source), $f_g[n]$ denotes glottal flow, g_v is gain of glottal pulses that provides the overall gain in the vocal tract, producing $g_g[n]$ signal, that is passed through vocal tract $f_v[n]$ producing $v_f[n]$. Finally, the signal is passed through oral cavity modeled by radiation loss $f_r[n]$ producing output signal $s[n]$.

18.6.4 Unvoiced Speech

When the excitation source in our corresponding model is a random white Gaussian noise, we consider that sound to be unvoiced. In the English alphabet “F”, “SH” are examples of unvoiced speech. To influence the speech sound, we have the following parameters in our speech production model:

- The mixture between voiced— v and unvoiced— u excitation (determined by v and u)
- The fundamental frequency (determined by $P(f)$)
- The spectral shaping (determined by $H(f)$)

The excitation source comes from the lungs through the larynx. The output of both generators is then added and fed into the box modelling the vocal tract and performing the spectral shaping with the transmission function $H(f)$. The emission characteristics of the lips is modelled by $R(f)$. Hence, the spectrum $S(f)$ of the speech signal is given as 18.5:

$$S(f) = (v \cdot P(f) + u \cdot N(f)) \cdot H(f) \cdot R(f) = X(f) \cdot H(f) \cdot R(f) \quad (18.5)$$

We can now transform the product of the spectral functions to a sum by taking the logarithm on both sides of Eq. 18.5 to produce Eq. 18.6:

$$\begin{aligned} \log(S(f)) &= \log(H(f) \cdot U(f)) \\ &= \log(H(f)) + \log(U(f)) \end{aligned} \quad (18.6)$$

Squaring yields 18.7:

$$\begin{aligned}\log(|S(f)|^2) &= \log(|H(f)|^2 \cdot |U(f)|^2) \\ &= \log(|H(f)|^2) + \log(|U(f)|^2)\end{aligned}\tag{18.7}$$

In the log-spectral domain we could now subtract the unwanted portion of the signal, if we knew $|U(f)|^2$ exactly. To get rid of the influence of $U(f)$, one would have to get rid of the "high-frequency" parts of the log-spectrum. This requires the use of low-pass filtering. The filtering can be done by transforming the log-spectrum back into the time-domain (in the following, \mathcal{FT}^{-1} denotes the inverse Fourier transform 18.8):

$$\begin{aligned}\hat{s}(d) &= \mathcal{FT}^{-1} \left\{ \log(|S(f)|^2) \right\} \\ &= \mathcal{FT}^{-1} \left\{ \log(|H(f)|^2) \right\} + \mathcal{FT}^{-1} \left\{ \log(|U(f)|^2) \right\}\end{aligned}\tag{18.8}$$

The excitation for the unvoiced speech is weighted by a gain factor g_n as shown in Fig. 18.12. It then goes through the vocal-tract and the lip radiation filter to generate unvoiced speech $s[n]$. For the unvoiced case, the transfer function of the vocal tract $H(f)$ is multiplied with the spectrum.

From Fig. 18.12 it can be seen that the source $u_n[n]$ is multiplied by $g_n[n]$. This is the input to the vocal-tract filter and generates $v_f[n]$ that produces the unvoiced speech $s[n]$. The unvoiced speech presented in the following equations:

$$\begin{aligned}v_f[n] &= (g_n u \otimes f_v)[n] \\ s[n] &= (f_r \otimes v_f)[n]\end{aligned}\tag{18.9}$$

In the next Fig. 18.13 the illustration of voiced signal and unvoiced signal is depicted:



Fig. 18.12 Unvoiced speech in the source excitation model

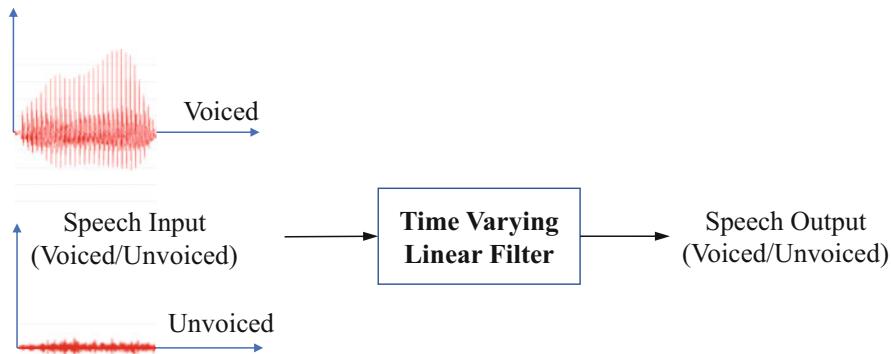


Fig. 18.13 Voiced and unvoiced speech example

18.7 Silence Regions

The speech production process involves generating voiced and unvoiced speech in succession, separated by what is called a silence region. During a silence region there is no excitation supplied to the vocal tract and hence no speech output. Silence regions are quite arbitrary and occur not only in speech. Silence plays an important role for speech:

- Silence is needed to produce speech properly (inhalation of air in breathing cycle)
- Silence as a semantic issue. This issue is discussed in the psycho-acoustics and hence is not elaborated here.

We will first look at the speech production. Silence is an integral part of the speech signals. Without the presence of silence region between voiced and unvoiced speech, the speech will not be intelligible. Further, the duration of silence along with other voiced or unvoiced speech is also an indicator of a certain category of sounds. A signal energy may be very low and close to silence, but its duration is important for perceiving it. Besides the acoustic means silence has a semantic importance for many applications. In medicine for instance if the heartbeat is silent this is a dangerous information. If there is silence during a speech that can also tell us something. These issues will be discussed later on. The computation of the envelop using the Hilbert transform maintains a representation of the signal. The envelop of the signal using the Hilbert transform sets a qualitative boundary around the silence. The silence intervals from the speech are removed using a threshold. One commonly takes for the threshold one fourth of a median of the envelop for removing speech silence, or pause, or clicking sounds from spoken speech. There is again no precise reason for doing so. A specific method for handling noisy speech is the pre-emphasizing approach. Here one can modify and extend an existing approach. In discourse analysis, speakers use brief absences of speech to mark the boundaries of prosodic units. Silence in speech can be due to hesitation, self-

correction—or a deliberate slowing of speech to clarify or aid the processing of ideas. In speech a silence interval is also called a pause where the noise is neglected. A method to detecting a pause is to use the property that noise has a lower variability than speech. Therefore one monitors the fluctuation of the spectral envelope. One way to do it is to compute the spectral frequencies. During voiced periods there occur large fluctuations. The detection of pauses analyses the difference between the fluctuation of the spectral envelope and the frequencies.

18.8 Glottis

A Latin word for glottis is “cavitas larynges intermedia”. The glottis is defined as the vocal folds and the opening between them. As the vocal folds vibrate, the resulting vibration produces pulses of air that we term as a quality to the speech, called voicing. The vibration is a component of vowels. The vocal tract that contains the vocal cord system takes a continuous input which is the excitation source and produces a periodic airflow as an output. In the real world, the technical analysis of the vocal tract system is modeled mostly applying linear signal model (e.g., 18.3) for a simple yet effective computation.

18.9 Lips

Lips serve for the articulation of sound and hence speech. The lips and the nasal radiation process put the final emphasis on the speech generation. The lips are in close connection to the teeth with which they cooperate. Lips serve for the articulation of sound and speech. The lips enable whistling and the performing of wind instruments. The lips and the nasal radiation process put the final emphasis on the speech generation. $F_r(z)$ is the z -transform of $f_r[n]$. The lip radiation filter is responsible for the speech sound which radiates through the lips. This is modeled as a single zero lying inside the unit circle. This filter can be expressed as an infinite product of the poles inside the unit circle. The z -transform $F_r(z)$ is defined in 18.10 as:

$$F_r(z) = 1 - a_r z^{-1} \quad (18.10)$$

18.10 Plosive Speech Source

During the impulse input or the plosive input, the glottis or the glottal flow generator, $F_g(z)$, has in general no glottal flow or it has no contribution in generating the unvoiced or the plosive speech. Therefore, the glottis is omitted. The glottis is defined as a combination of the vocal folds and the space in between the folds. We assume that the excitation source $u_s[n]$ is generated from lungs. It is weighted by the gain g_i and then goes through the vocal-tract filter and the lip, adding radiation effect to generate speech such as “B”—voiced stop, or corresponding unvoiced stop “P”. In Fig. 18.13, g_i is a impulsive source. It is a volume controller and produces $v_f[n]$ when it goes through the vocal-tract filter $f_v[n]$. Finally, $s[n]$ is the response of $v_f[n]$ at the lip depicted with radiation filter $f_r[n]$. Hence, the following notation is used $U_s(z)$ and $S(z)$ as the z-transforms of respectively $u_s[n]$ and $s[n]$.

Plosive sound is produced by a brief region of silence, followed by a region of the voiced or the unvoiced speech. A plosive example (silence + unvoiced) is the sound corresponding to /t/ in “pat”. Another example (silence + voiced) is the /b/ in “boot”). In Fig. 18.14, the sound is modeled by a pulse and $g_i[n]$ is indicating the gain or weight that one sometimes denotes as volume of the sound. Equation 18.11 gives the mathematical formulation of this sound. The sign “ \otimes ” is the symbol for the convolution operation.

$$\begin{aligned} v_f[n] &= g_i[n]u_s[n] \otimes f_v[n] \\ s[n] &= f_r[n] \otimes v_f[n] \end{aligned} \quad (18.11)$$

18.11 Vocal-Tract

A significant importance is given to the vocal-tract in the speech production modeling next. The vocal-tract is an acoustical tube shown in Fig. 18.15. It can be seen that the vocal-tract has different cross sectional areas which are denoted by A_1, A_2, \dots, A_6 as shown in Fig. 18.15. There is a relationship with the non-uniform

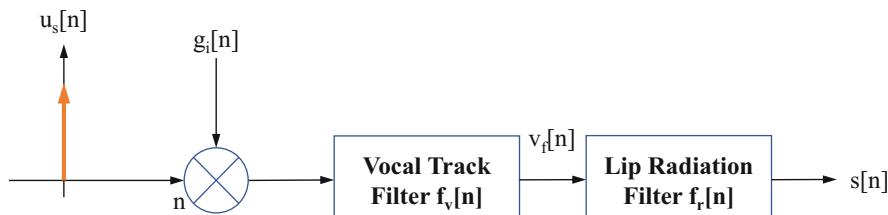


Fig. 18.14 Plosive sound generation in source excitation speech model

vocal-tract cross-sectional area and the discrete time sampling rate selection during C/D conversion of the analog speech waveform. The discrete time speech production model assumes the vocal-tract is a concatenation of number of tubes. In Fig. 18.15 it can be seen that the vocal-tract is not uniform. This introduces discontinuities, that are handled by derived in Part I via transfer function of the vocal-tract system that is repeated here for clarity 18.12.

$$f_v(z) = \frac{1}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (18.12)$$

The presented equation describes an all pole filter because a polynomial of order p appears in denominator, and the polynomial of order p has that many roots that are called poles and are determined based on a_i coefficients, while the numerator is 1.

The length of the vocal-tract for a typical adult is about 17.7 cm and the speed of the sound in the air is assumed to be 340m/s.¹ Hence the vocal tract is the center of the human speech production. In Fig. 18.15 a vocal tract is shown as a concatenation of loss-less acoustical tubes. Of interest is that we see different cross-sectional areas A_i . From the figure it can be observed that the vocal-tract has different cross sectional areas denoted by A_1, A_2, \dots, A_6 . If the vocal-tract produces different kinds of sounds that are dependent on its shape. However, its shape, and hence the shape of the vocal-tract changes slowly (e.g., due to mass of the tissue and limited amount of exerted force). From glottis to lips, it is considered as the vocal-tract filter $F_v(z)$ and one models it by using an all-pole filter shown in Eq. 18.12. The $F_v(z)$ is a z -transform of $f_v[n]$. This filter is unique for a particular speaker and also it is unique for each speech sound spoken by a speaker. We can see that Eq. 18.8 shows an all pole filter because the denominator there is indicated by a_i coefficients where for $i = 1, 2, \dots, p$, and p equals to number of poles. The sounds generation is modeled by overall production as depicted in block diagram illustrated in Fig. 18.16:

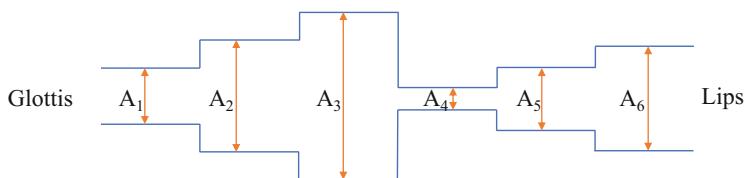


Fig. 18.15 Discrete Vocal Tract model of a number of uniform tubes

¹ The speed of sound varies depending on the medium, for example, sound waves move faster through water than through air. Also the speed of sound is dependent on the temperature and altitude of the location as well as other factors that we will not discuss here.

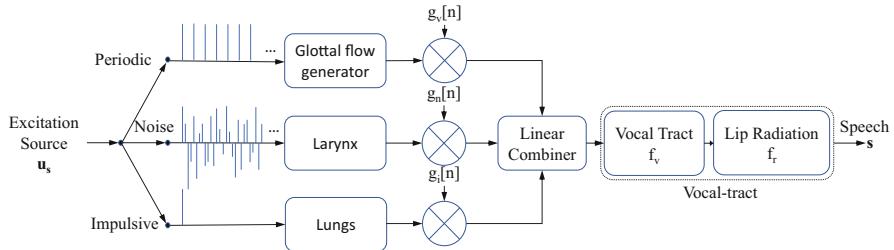


Fig. 18.16 Source excitation speech model

From the figure it can be seen that each source is multiplied by a gain factor. The gain varies and it changes according to the speaker and the speech generated by the speaker.

18.12 Parametric and Non-parametric Models

In this section we discuss auto-regressive (AR) parametric signal modeling. This model can be stochastic or deterministic. The source excitation model is in fact the stochastic type *AR* model. The model parameters are not known and they have to be determined. We first introduce the notion of parametric signal modeling. The model parameters have to be solved based on the model. We introduce the notion of parametric signal modeling in contrast to non-parametric modeling. A non-stationary signal is generally analyzed in a small segment. This small section can also be modeled by using a parametric signal model or by a non-parametric signal model. In a parametric signal modeling this small segment is modeled by a number of parameters. These parameters may change from segment to segment. This generally happens when a non-stationary signal, such as one from speech, is modeled. In this modeling, most often a complicated process such as a speech signal can be represented by a smaller number of parameters than the actual samples in the signal. The parameters capture changes on dynamics of the signal. That means the signal parameters reflect the changes of the signal. The reduction of the parameters often requires an approximation, estimation and also some constraints or some additional information. A common approximation is when the system is driven by some known input. The input is most often assumed as a unit sample signal or white Gaussian noise. On the other hand, in the non-parametric signal modeling, the signal is most often characterized by the measurements of the frequency response and this may contain a large number of frequencies. The optimal spectrum of the excitation in the parametric case will be different from that in the non-parametric case. This is principally because the parametric model combines the information available from all frequencies in only a few parameters. In a direct non-parametric frequency response measurement there is no relation between the measurements at

the various frequencies and therefore the excitation should be designed to achieve a predefined accuracy in the frequency bands of interest. An example is maximizing the absolute or relative accuracy of the measurements. In a parametric approach the energy will be concentrated on the frequencies where it contributes most to the knowledge about the model parameters. Some important factors to be considered are:

- Model type
- Model order
- Approach to estimate model parameters

We specify parametric model by:

- The AR type of model used
- Model order

The basic parametric modeling is based on the auto-regressive (AR) moving average (MA) parametric model known as ARMA. It incorporates a moving average autoregressive filter. The basic parametric modeling, namely the auto-regressive model, the moving average model and the ARMA model use mostly the least squares criteria in order to estimate the model parameters. The AR modeling is commonly used for the signal modeling because it is easily tractable for parameter estimation. First we make a very general description.

Figure 18.17 displays the parametric modeling approach to ARMA. The individual section of the ARMA filter are presented in Fig. 18.18. The upper half describes MA and the lower half describes AR.

Suppose the speech $s[n]$ is a response of a system $h[n]$ which has the excitation $u[n]$. This means $u[n]$ is the input, $s[n]$ is the output and $h[n]$ is the system. Then one assumes that $u[n]$ is white Gaussian noise and $h[n]$ is modeled by the ARMA model defined in Fig. 18.18. The model has two parts: One is for the AR model with order p and another part is for the MA model with order q . For the white noise as input u , the system h generates s . We describe this now in the z-domain where $S(z)$, $U(z)$ and $H(z)$ are the z-domain representation of $s[n]$, $u[n]$ and $h[n]$.

The expansions of $B(z)$ and $A(z)$ are given in Eqs. 18.13–18.14.

$$B(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q} \quad (18.13)$$

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p} \quad (18.14)$$

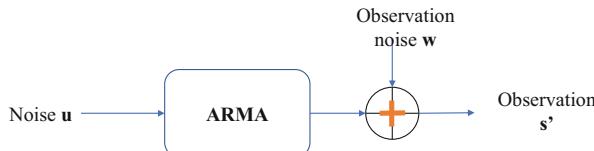


Fig. 18.17 Parametric speech source modeling using ARMA model

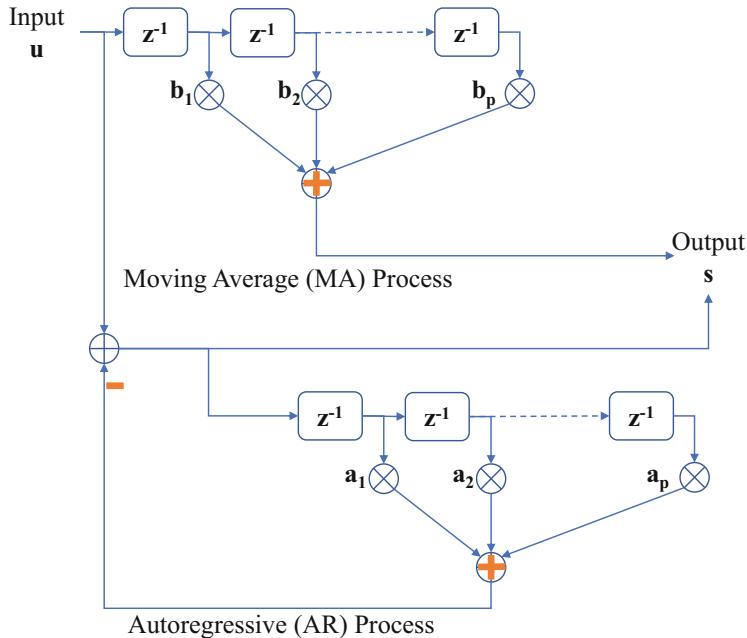


Fig. 18.18 Moving average and regressive processes

The input-output relation is shown in Eq. 18.15

$$s[n] + \sum_{i=1}^p a_i u[n-i] = u[n] + \sum_{i=1}^q b_i u[n-i] \quad (18.15)$$

The speech signal changes with time and its different phonemes have different characteristics as expressed in their waveform. The phoneme is the fundamental unit of the sound. In Eq. 18.16, we see that the speech signal $s[n]$ is a linear combination of its past p samples and the excitation $u[n]$ multiplied by a weight:

$$s[n] = \sum_{i=1}^p a_i s[n-i] + g_s u[n] \quad (18.16)$$

Taking the z-transform one finally arrives at:

$$\frac{S(z)}{U(z)} = \frac{g_s}{1 - \sum_{i=1}^p a_i z^{-i}} = \frac{g_s}{A(z)} \quad (18.17)$$

where, a new factor g_s was introduced.

18.13 Formants

There are two definitions for a formant:

- (i) The spectral peak of the sound spectrum.
- (ii) A range of frequencies of a complex sound for which there is an absolute or relative maximum.

We will use the first interpretation. Formants denote the distinguishing or meaningful frequency components of human speech. In speech science a formant is also sometimes used to mean an acoustic resonance of the human vocal tract. Therefore a formant can denote either a resonance as well as the spectral maximum of the resonance. Formants are often measured as amplitude peaks in the frequency spectrum of the sound. They can be estimated by a spectrogram or a spectrum analyzer of the vocal tract resonances. In vowels spoken with a high fundamental frequency, as in a female or child voice, the frequency of the resonance may lie between the widely-spaced harmonics and hence no corresponding peak is visible.

18.14 Strong Noise

One of the noise types discussed above was strong noise. We restricted them such that we could consider them as outliers and handled them by removing the outliers. After the outlier removal there are two possibilities:

- The speech is not affected by the removal. In particular, the understanding is not disturbed because in the removed intervals no speech took place, i.e. it was a region of silence.
- Some part of the speech is removed. As a consequence, the speech is incompletely or not at all delivered which may affect the understanding.

In the first case one can proceed in the ordinary way. The second case is more difficult. For handling it, the system has to reconstruct the missing part. In addition the speaker has to be informed that something is missing. To do this one has to have additional capability provided by the system that would generate a feedback message from the receiver (machine) to the user (a human) indicating the inability to understand the provided speech, e.g.: “Repeat the message”. The realization of the feedback could provide a solution. A strong noise case requires methods for detecting them. This will be done by using probabilities. A brief introduction is provided in the next chapter.

18.15 Background Information

Development of speech recognition (SR) started in the 1950s and 1960s. Initially SR systems could understand just numbers. Among the first commercial systems mentioned was the “Audrey” system designed in 1952 by Bell Laboratories run on the “Shoebox” machine by IBM in 1962. Those initial systems utilized dynamic programming technique called Dynamic Time Warping to perform basic pattern matching.

Speech recognition technology made a major step forward in the 1970s with introduction of Hidden Markov Models. For further reading on speech we recommend Deller et al. (1993). The variations in stochastic speech processes are discussed in Vaseghi (2008). The ARMA denotes the kinds of stationary stochastic processes. In the statistical analysis of time series, autoregressive-moving-average (ARMA) models provide a description of stochastic processes in terms of two polynomials, one for the auto-regression and the second for the moving average. The ARMA model was described in the Box et al. (1994). Today there are several commercial speech recognition systems. Prominent commercial examples are: Amazon “Echo”, Microsoft “Cortana”, Google “Home”, Apple “Siri”. More prominent open source examples are CMU Sphinx <http://www.speech.cs.cmu.edu/sphinx/doc/Sphinx.html>, and Kaldi <https://www.kaldi-asr.org>.

Formants have been introduced in Fant (1960). The second definition of formants can be found in the American Standards Secretariat (1994). Neural nets for speech recognition are studied in Harb and Husseiny (2000). Some sources are available in Quatieri (2007), Hasegawa-Johnson (2005) and readers are encouraged to take a look at these reference.

18.16 Exercises

Exercise 1 Give two examples, one where the autoregressive function is deterministic and one where it is stochastic.

Exercise 2 Describe precisely in numerical terms where strong noise occur.

Exercise 3 Describe precisely a situation where plosive occurs.

References

[Deller1993] Deller, J.J, Proakis, J, Hansen (1993). Discrete-Time Processing of Speech Signals. Macmillan Publishing.

[Box1994] Box, George, Jenkins, Gwilym M., Reinsel, Gregory C. (1994). Time Series Analysis: Forecasting and Control (Third ed.). Prentice-Hall.

[Quatieri2007] Quatieri, T.F. (2007). Discrete-Time Speech Signal Processing. Prentice Hall.

- [Vaseghi2008]** Vaseghi, S.V. (2008): *Multimedia Signal Processing: Theory in Speech, Music and Communications*. Wiley, USA.
- [Hasegawa2005]** Hasegawa-Johnson, M. (2005). Acoustic Features. <http://www.ifp.uiuc.edu>.
- [Fant1960]** Fant, G. (1960). *Acoustic Theory of Speech Production*. Mouton & Co, Netherlands.
- [ASA1994]** Acoustical Society of America, (1994). ANSI S1.1-1994 (R2004) American National Standard Acoustical Terminology.
- [Harb2000]** Harb, H., A. H. Husseiny (2000). Isolated words recognition using neural networks. International Conference on Electronics, Circuits and Systems, 1:349–351.

Chapter 19

Noisy Speech



Overview

In order to deal with signals and in particular speech signals in a computational way one needs a model of the system according to which computations can be performed. There are different kinds of models depending on the purpose of the intended computation goal. We will discuss some of them in the following sections.

19.1 Introduction

Colors, e.g. White, Blue, Purple, Violet etc., are used to denote certain properties of noise and of their effects to a human ear. Hence they are special to signal processes of speech. Human can identify different colors of noise due to its differing properties. The colors are used to name various types of noises due to the psycho-acoustic phenomena triggered by noise.

We considered noise in general and will add some aspects specific to speech and other signals. Of particular importance is how much noise can disrupt the understanding of the signals.

19.2 Colored Noise

The colored noise modeled by the Gaussian process is depicted in the picture presented here and is treated by applying adaptive filtering.

The below presented equation is a depiction of Gaussian noise, $p(x)$, with mean μ and variance σ^2

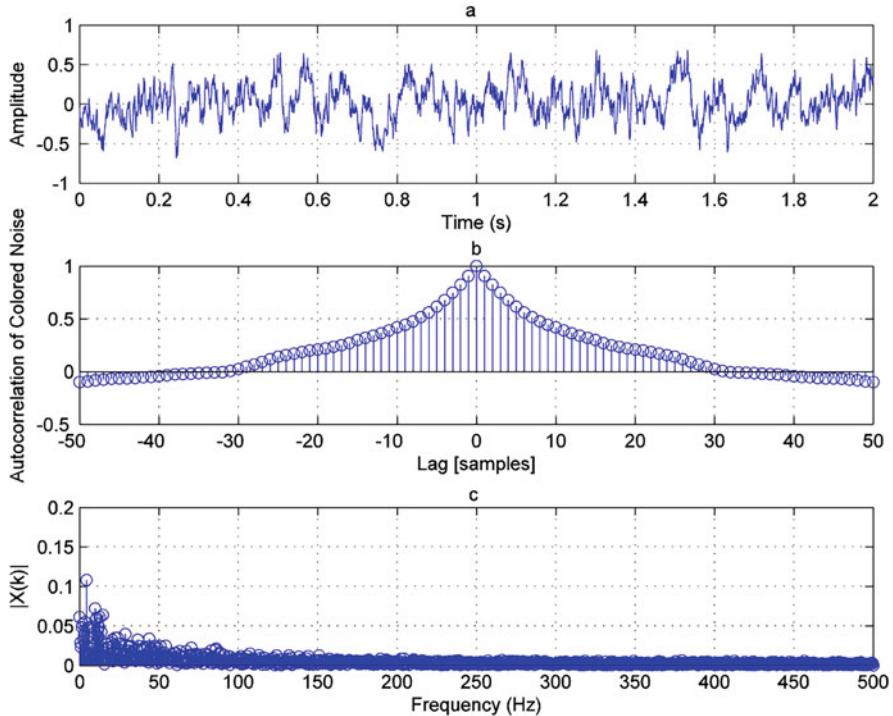


Fig. 19.1 Gaussian model for colored noise. (a) Colored noise. (b) Autocorrelation of colored noise. (c) Frequency information of colored noise

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)} \quad (19.1)$$

The noise model shown in Fig. 19.1 and with Eq. 19.1 is an auto-regressive (AR) model and its parameters are obtained by the linear prediction, namely the Yule-Walker approach. In Eq. 19.2, the noise $d[n]$ is a linear combination of past i samples and β_i coefficients and a disturbance $w[n]$. This is assumed to be a white noise and it is weighted by g_b .

$$d[n] = \sum_{i=1}^q \beta_i d[n-i] + g_b w[n] \quad (19.2)$$

For treating this noise, one approach is for the signal to be first divided into sub-bands using a cosine modulated quadrature mirror filter bank (QMF) and then the noise is minimized for each sub-band by a spectral minimization technique. The signal is enhanced in each band by a Kalman filter. In this noise reduction,

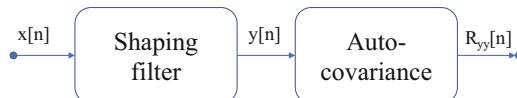


Fig. 19.2 Shaping filter where $y[n] = ay[n - 1] + (1 - a)x[n]$

noise is varying in each sub band. Due to the fact that speech is corrupted by noise, it is desirable to categorise various types of noise. Colored noise can be generated by passing the white noise through shaping filter. The shaping filter is a dynamic filter, usually a low pass filter. The response of the colored noise can be varied by adjusting the parameters of the shaping filter. The low pass filter can be implemented in various ways. Specifically, the Matlab's "FILTER" function is used in this simulation (Fig. 19.2).

19.2.1 Additional Types of Colored Noise

The practice of naming kinds of noise after colors started with "white noise", a signal where the spectrum has equal power within any equal interval of frequencies. Some of those names have standard definitions in specific disciplines, while others are very informal and partially inconsistent. Some are discussed now.

- (i) **White** noise which has equal strength over a linear scale of frequencies but is not perceived as being equally loud in the human ear. Other color names, like:
- (ii) **Grey** noise is variation of white noise. It is randomly distributed. It is used for describing loudness of speech. This is not as for standard white noise where we have an equal-loudness contour.
- (iii) **Red** noise is often used as a reference signal in audio technique. Sometimes it is termed as pink noise. It has equal power in the frequency range from 40 to 60 Hz as in the band from ca. 4000–6000 Hz. Every octave contains the same amount of energy.
- (iv) **Blue** noise is a name given to noise with other spectral profiles; often (but not always) in reference to the color of light with similar spectra. Its power density increases 3 dB per octave with increasing frequency (density proportional to f) over a finite frequency range.
- (v) **Violet** noise power density increases 6 dB per octave with increasing frequency. Its density is proportional to f^2 over a finite frequency range. Sometimes it is called as differentiated white noise, due to its being the result of differentiation of a white noise signal.
- (vi) **Pink** noise. While white noise has a consistent strength across various frequencies, pink noise has more variation. The lower frequencies in pink noise are louder and have more power than the higher frequencies that has equal power per octave. Examples of pure pink noise in nature include:

- Leaves rustling on a tree in the wind
- Waves on the coastline of a ocean
- Steady falling rain

Since there are an infinite number of logarithmic bands at both the low frequency and high frequency bands of the spectrum, any finite energy spectrum must have less energy than pink noise at both ends. Red noise is the only power-law spectral density that has this property: all steeper power-law spectra are finite if integrated to the high-frequency end, and all flatter power-law spectra are finite if integrated closer to low frequency band. The presented colors are of technical character and refer to properties of the power spectrum.

19.3 Poisson Processes and Shots

The process that follows the probability of a given number of events occurring in a fixed interval of time or space, and if these events occur with a known constant mean rate and independently of the time since the last event, is called Poisson Process. Formally, a stochastic process following density function described by a discrete random variable X is a Poisson process if it has random variable x and a parameter λ , where $\lambda > 0$, and it is described by Eq. 19.3 indexed by k :

$$f(k, \lambda) = \frac{\lambda^k}{k!} e^{-\lambda} \quad \text{for } k = 0, 1, 2, \dots \quad (19.3)$$

In Eq. 19.4 the function $\delta(t)$ is an impulse waveform that appears at random times with a Poisson distribution with amplitude a_i as indicated by the following equation:

$$X(t) = \sum_i a_i \delta(t - t_i) \quad (19.4)$$

Then, in a given time interval (t_1, t_2) we denote the number of occurrences in this interval by $n(t_1, t_2)$. To be precise, $n(t_1, t_2)$ is an integer valued random variable. For two finite non-overlapping intervals (t_1, t_2) and (t_3, t_4) the random variables $n(t_1, t_2)$ and $n(t_3, t_4)$ are independent. For a unit interval the average number of events is λ . For an interval of length t the probability that k events take place in the interval is:

$$Prob_\lambda(t, k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \quad (19.5)$$

Those events are distributed in a Poisson manner with parameter λt . For the integer valued random variable $n(t_1, t_3)$ we have:

$$E[n(t_1, t_3)] = \lambda(t_1 - t_3) \quad (19.6)$$

and

$$f(x) = P(X = n) = \frac{t^n}{n!} e^{-t} \quad (19.7)$$

$$F(n) = \sum_{k=0}^{n!} f(k) \quad n = 0, 1, 2, \dots \quad (19.8)$$

For example, if the events occur on the average 2 times per minute, and we are interested in the number of events occurring in a 20 minute interval, then the model using Poisson distributions is written in Eq. 19.3 with $\lambda = 20 \times 2 = 40$.

A shot is an interval that satisfies the conditions to be small and randomly distributed. Now we model noisy speech by a Poisson distribution. For this reason, a Poisson process is called a counting process. Here the Poisson process is used to interpret $N(t)$ as the number of arrivals of tasks (or jobs, or noise) to a system by a time t . If the time interval denoted by Δn , where $\Delta n \ll 1$, and the shots may not occur in every interval, then the probable events can be described by Eq. 19.9 presented below:

$$V_n = \begin{cases} 0, & \text{if no impulse occurs in the time interval } n\Delta n < (n+1)\Delta n \\ 1, & \text{if an impulse occurs in the time interval } n\Delta n < (n+1)\Delta n \end{cases} \quad (19.9)$$

If there is only one event in each interval then we can describe the probabilities of the V_n by Eqs. 19.10 and 19.11. The pulse $p(t)$ is confined to $-\Delta t \leq t \leq \Delta t$ and The impulse response $h(\tau)$ is confined to $0 \leq \tau \leq T$ hence $y(t)$ vanishes outside the interval $t_0 - \Delta t \leq t \leq t_0 + T + \Delta t$.

$$P(V_n = 0) = \exp(-\lambda\Delta n) \approx 1 - \lambda\Delta n \quad (19.10)$$

$$P(V_n = 1) = \lambda\Delta n \exp(-\lambda\Delta n) \approx \lambda\Delta n \quad (19.11)$$

For more events one has to be generalize the derivation. If the Δt is infinitesimally small then the summation is converted into an integral and the expected value of the shot noise process obtained by Eq. 19.12.

$$E[Y(t)] = \lambda \int_{-\infty}^{\infty} h(\tau) d\tau \quad (19.12)$$

The variance of the shot noise is shown in Eq. 19.13.

$$\sigma_y^2 = \lambda \int_{-\infty}^{\infty} h^2(\tau) d\tau \quad (19.13)$$

19.4 Matched Filters

Suppose the input signal $s_i(t)$ is defined within a finite time t . One can estimate the impulse of the optimum linear filter $h_{opt}(t)$ by running the signal backwards in time from the instant t_n at which the maximum signal to noise ratio (SNR) has occurred. This type of filter is generally called the matched filter. The matched filter is concerned with strong noise and is supposed to detect the shot noise. The optimum filter can be developed by following mathematical formulation shown in Eq. 19.14.

$$h_{opt} = \frac{2k}{N_0} S_t(t_n + \tau) \quad (19.14)$$

We return to a little more general situation. Suppose the observed signal x is noisy and the desired signal s is corrupted by the noise n , then we have $x = s + n$. If one models this event using an auto-regressive (AR) process and one can then predict s perfectly by \hat{s} using the linear prediction (LP) analysis then $s = \hat{s}$. If there is no noise then $x = s$. In the prediction analysis an exact measurement is not possible and there are some errors e which is approximated as white noise. Thus we assume the noise is approximated by some impulse signal and h_{opt} is the time reversed version of the impulse signal. The matched filter has to be at least as wide as the signal spectrum otherwise it will reject the signal and reduce the SNR. In addition, the extreme frequencies of the signal must be emphasized. When the noise is strong (see below) the associated frequencies must be deemphasized. The output of this matched filtering to our strong hybrid noisy signal is shown in Fig. 19.3 presented below.

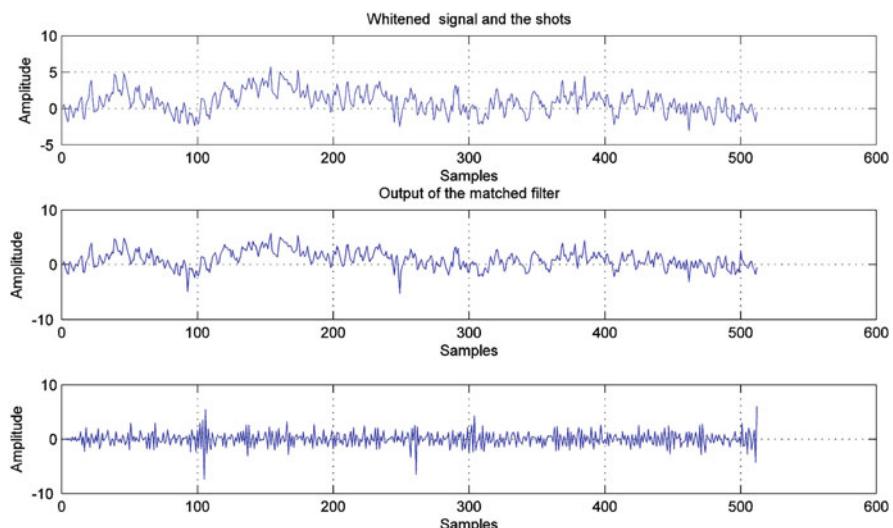


Fig. 19.3 Output of matched filtering

In this figure one sees the outputs of different samples for matched filtering. We want the filter that yields the highest signal-to-noise ratio (SNR) at its output with minimal alteration of the original signal. This is done by matched filtering.

19.5 Shot Noise

The shot noise occur when SNR is at its maximum. For studying this type of noise some concepts are needed that are illustrated first. Below the matched filter is briefly introduced for the white noise and colored noise. For white noise $n(t)$ with spectral weight $N_0/2$: Suppose $x(t)$ as the mixture of the signal $s(t)$ and the white noise $n(t)$, that is:

$$x(t) = s(t) + n(t) \quad (19.15)$$

is the input to the system $h(t)$ such that the output is $y(t)$ as shown in Fig. 19.4

The signal-to-noise ratio for the signal y_s and the noise y_n is:

$$SNR = \frac{y_s^2(t)}{E[y_n^2(t)]} = \frac{\left[\int_0^t s(u)h(t-u)du \right]^2}{E \left[\int_0^t n(u)h(t-u)du \right]^2} \quad (19.16)$$

The noisy signal y_n in denominator of Eq. 19.17 is:

$$\begin{aligned} E[y_n^2(t)] &= E \left[\left\{ \int_0^t n(u)h(t-u)du \right\} \left\{ \int_0^t n(v)h(t-v)dv \right\} \right] \\ &= \int_0^t \int_0^t E \left\{ n(u)n(v) \right\} h(t-u)h(t-v) du dv \\ &= \int_0^t \int_0^t \frac{N_0}{2} \delta(u-v) h(t-u)h(t-v) du dv \end{aligned}$$

$$E[y_n^2(t)] = \frac{N_0}{2} \int_0^t h^2(t-u) du = \frac{N_0}{2} \int_0^t h^2(t-v) dv \quad (19.17)$$

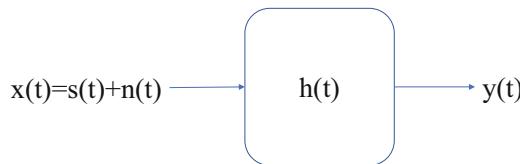


Fig. 19.4 Additive noise corrupted signal

From Eq. 19.17, one obtains optimized SNR in Eq. 19.18:

$$SNR = \frac{y_s^2(t)}{E[y_n^2(t)]} = \frac{\left[\int_0^t s(u)h(t-u)du \right]^2}{\frac{N_0}{2} \int_0^t h^2(t-u)du} \quad (19.18)$$

If $s(t)$ is of finite duration T , then SNR is maximized by setting $t = T$, that is 19.19:

$$SNR = \frac{\int_0^T s^2(u)du}{\frac{N_0}{2}} = \frac{2\varepsilon_s}{N_0} \quad (19.19)$$

The matched filter works as the maximum SNR filter when its impulse response matches the noise impulses. Thus the matched filter impulse response is the flipped version of the signal as shown in Eq. 19.20.

$$h(T-u) = c \cdot s(u) \text{ or } h(u) = c \cdot s(T-u) \quad (19.20)$$

The output of the matched filter for the finite signal length T is the inner product of the signal and the input as shown in Eq. 19.21.

$$\begin{aligned} y(T) &= \int_0^T h(u)x(T-u)du = \int_0^T c \cdot x(T-u)du \\ &= c \int_0^T s(u)x(u)du = x \cdot s \end{aligned} \quad (19.21)$$

Taking the Fourier transform of the matched filter impulse response of the Eq. 19.22

$$\begin{aligned} H(f) &= \int_0^T h(u)e^{-j2\pi f u}du = c \int_0^T s(T-u)e^{-j2\pi f u}du \\ &= c \int_T^0 S(x)e^{-j2\pi f(T-x)}(-dx) = | \text{ for } x = T-u | \\ &= ce^{-j2\pi f T} \int_T^0 S(x)e^{j2\pi x}dx = ce^{-j2\pi f T}[S(f)]^* \end{aligned} \quad (19.22)$$

The magnitude of the matched response is just a scaled version of signal as seen in Eq. 19.23

$$|H(f)| = c|S(f)| \quad (19.23)$$

In order to detect strong noises we regard them as shots. Hence we have to detect shots. We start with some general considerations. For this we first maximize the signal to noise ratio (SNR). This will be achieved by a certain filter that is called matched filter. If a signal f is mixed with a noise v then the resulting x is in 19.24:

$$x(t) = f(t) + v(t) \quad (19.24)$$

The signal $f(t)$ is known in principle but not its location—which is what we want to detect. For this we consider some filter a linear filter with the response $h(t)$. This results in the Equation $y(t) = x(t) \times h(t)$. Expanding this gives the following 19.25:

$$y(t) = \int_{-\infty}^{+\infty} x(t-u)h(u)du \quad (19.25)$$

Linearity allows splitting of y as in 19.26:

$$y(t) = y_f + y_v \quad (19.26)$$

By performing Fourier transformation on original signals, $F(\omega)$, $S(\omega)$, and $H(\omega)$ of v and h the following 19.27, is obtained:

$$y(t) = \int_{-\infty}^{+\infty} F(\omega)H(\omega)e^{-j\omega t}d\omega \quad (19.27)$$

In order to detect the shots we insert a filter between the input and the matched filter. This is done in such a way that the transfer function of the inserted filter is working.

19.6 Background Information

In general, a matched filter is intended to optimize the signal-to-noise relation. For this purpose the filter is used to detect the existence of some amplitude even if noise is present. This is described in North (1943). For Fig. 19.2 on shaped filters see Gaussian Waves (2020).

Suggested

The following example shows how to identify a keywords in noisy speech using a deep learning network. This example uses Audio Toolbox and Deep Learning Toolbox. For further information please see “Keyword Spotting in Noise Using MFCC and LSTM Networks”, <https://www.mathworks.com/help/audio/ug/keyword-spotting-in-noise-using-mfcc-and-lstm-networks.html>

Voice Activity Detection in Noise Using Deep Learning—This example shows how to detect regions of speech in a low signal-to-noise environment

using deep learning. This example can be accessed using this link: https://www.mathworks.com/help/audio/ug/voice-activity-detection-in-noise-using-deep-learning.html?searchHighlight=voice%20activity%20detection&s_tid=srchtitle_voice%2520activity%2520detection_4

19.7 Exercises

Exercise 1 Give an example of shots in physics.

Exercise 2 Provide a proof for Eq. 19.22.

References

- [North1943] North D. O., Analysis of the factors which determine signal/noise discrimination in radar. In: Report PPR-6C, RCA Laboratories, Princeton, NJ.
- [GaussianWaves2020] www.gaussianwave.com, 2020

Chapter 20

Aspects of Human Hearing



Overview

In a naive view it seems that spoken and written words are just different ways to represent the same things. That means they are equivalent and what can be expressed in one form can also be expressed in the other form. This neglects the fact that both forms are based on quite different inherent elements. Written words are represented by letters from a finite alphabet. Spoken words are represented by signals, more precisely by stochastic processes of signals. Signal processes have a greater and broader scope than simply sequences of letters. For example, the pitch of a utterance is a psychoacoustic property of the produced signal that is characterized by the frequency, loudness, intensity or amplitude of the acoustic sound.

The psychoacoustic phenomena reflect to properties of stochastic processes. These phenomena are known by the humans who hear the speech, with unknown properties of signal processes, and how these properties can be computed. The speech sounds arrive and they can be considered a random process. The speech signal exhibits variations both in the spectral and temporal domain. The human ear and brain together analyze the frequency of the speech. The outer ears accept speech sound pressure waves and send this through the middle ear to the inner ear. This finally transmits the sound that is pre-processed by the human ear to the brain. Thus we hear and recognize the speech.

Spoken language contains more information than a written text. If one expresses statements in spoken language one can easily hear that this statement, for example, affirmative or ironically intended, and the degree of expressed affirmation or intention. This is impossible to decipher from the written text. In order to express this one has to reformulate it in such a way that the intention is expressed explicitly. The intended information is hidden in the speaking style and contained in the wave forms obtained by the receiver.

In the same way as one obtains speech information as the words (without knowing their meaning) one extracts other information about speech that is concerned about psychoacoustic elements. Again, these elements are purely syntactic and are

not associated with any meaning. What we are interested are to find out what the syntactic elements are and how one can extract them.

20.1 Human Ear

This section complements the earlier analysis of the human speech generation. The human ear receives waves from the speaker. These waves are transformed into digital sequences. Then the ear starts a complex process of decoding speech that finally ends in the brain of the receiver. This process is roughly indicated in Fig. 20.1. In order to automate this reception it is useful to study the ear and the generation process. It turns out that the ear is a highly sophisticated organ that can perform quite difficult mathematical operations. The area of Automatic Speech Recognition (ASR) tries to mimic these operations or to replace them by simpler ones with the same effect.

The central technical concepts presented here are scales and filter banks. The scales are used to measure digital signals. The filters are used to define transformations, for instance, from one scale to another. We begin description by first looking at the human ear depicted in Fig. 20.1. It contains the major elements of the ear.

In speech perception, the sound pressure wave at the outer ear of the listeners is converted into neurological pulses in the middle and inner ear. These pulses are later interpreted in the auditory cortex of the brain.

The human ear is able to perform very useful signal processing on incoming signals. In the peripheral auditory system, the input is an acoustic signal and the output is a collection of neural spikes that enter the brain. The three components of the peripheral auditory system (outer ear, middle ear, and inner ear) are shown in Fig. 20.1.

20.2 Human Auditory System

The sound travels from the outer ear through the ear channel to the tympanic membrane. The vibrations are transmitted to the hair cells connected to the fluid filled passage in the inner ear. The vibrations generate signals which are carried out to the brain for the sound interpretation through the auditory or acoustic nerve. There are about 16000–20000 hair cells along the length of the cochlea in 4 different rows. There is only one row in which the inner hair cells are attached to nerves and the rest of the rows are attached to outer cells. These hair cells in the cochlea play a significant role in the properties of the sound as, for example, pitch, loudness and how these properties stimulate the hair cells and send a signal to the brain.

The idea of the fluid filled cavities is a movement of the bones caused by a vibration wave in the fluid which stimulates the microscopic hair cells connected to the nerve. Moving back and forth, the hair cells connected to the cavities in the

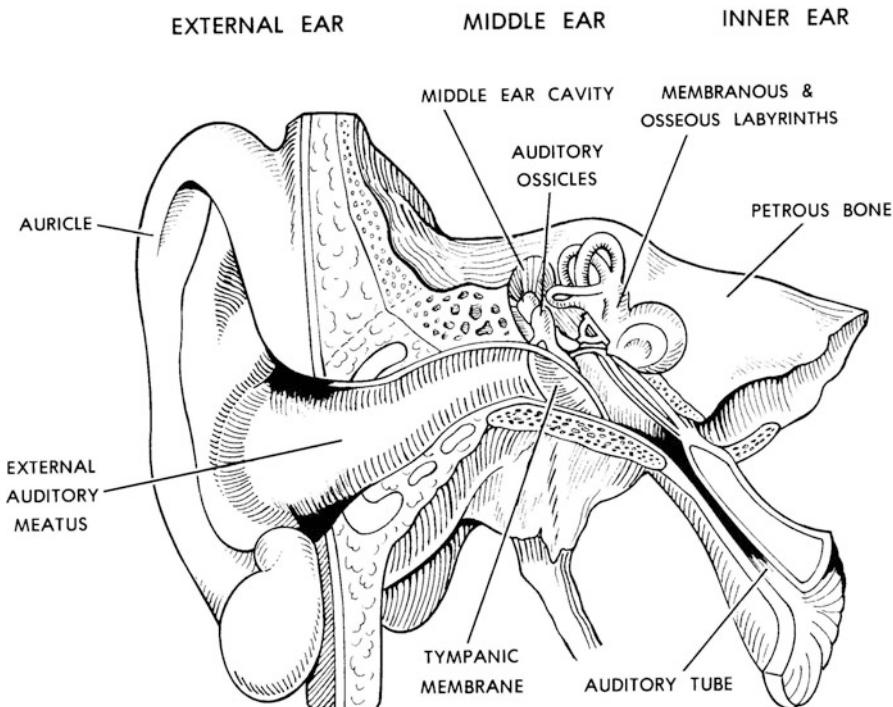


Fig. 20.1 Figure depicting human ear. <https://commons.wikimedia.org/w/index.php?curid=1678037>

cochlea fire electrical signals or impulses that are carried out to the brain through the auditory or acoustic nerves to the brain as an interpretation of the sound. The vibration of the hair cells at different rates helps the brain to interpret the sound frequency. The hair cells vibration in the Vestibular is the sensation of the position of the body and the head for a possible balance towards the hearing.

The auditory nerve sends the electrical impulses from the cochlea through the semicircular canals to the brain and thus relates to the auditory system and the message coding in the brain. The human ear is subjective to its response to different frequencies. This characteristic is technically achieved using many different types of scales. The peripheral auditory system acts as a frequency analyzer. The basilar membrane of the inner ear plays a significant role to locate and characterize the frequency. An auditory filter can measure the neural tuning curve and neural impulse responses. The auditory filter is mapped on the critical band to represent the frequency resolution of the auditory system.

20.3 Critical Bands and Scales

In psychoacoustics the concept of critical bands, introduced by Harvey Fletcher in 1933 and refined in 1940, describes the frequency bandwidth of the “auditory filter” created by the cochlea. Roughly, the critical band is the band of audio frequencies within which a second tone will interfere with the perception of the first tone by auditory masking. The central technical concepts in this section are scales and filter banks. A scale is something ordinary as we think as when referring to temperature in Celsius and Fahrenheit, which are two different temperature scales. The choice of scales depends on the application. For example we may ask “what is 0.13 feet in centimeters” or “what is 63 centimeters in feet”?

Filter banks are generalizations of filters that allows several filters to be applied simultaneously. This requires comparing filters for which scales are used. First we start with a number of filters in a bank of filters. Scale in the context of filters are mostly referring to the frequency bandwidth of the “auditory filter” in human cochlea. One purpose of scales is to provide steps that correspond to equal perceptual intervals.

Mel, Bark and Erb scales are discussed in this section. These are measured in logarithmic scales. The scales give different intervals for the linear frequencies. These scales influence the way in which the speech is recognized. They have also an influence on psychoacoustics as discussed in Sect. 20.7. Scales are connected to several numerical properties and functions. This has so far no rigorous mathematical foundations that clarify this. Therefore one cannot get a uniform mathematical way to describe them. Hence, one relies on experiments in order to obtain an approximation of the true formula.

20.3.1 Mel Scale

The Mel scale (measured in Mel) is a non-linear scaling that is used to describe frequency characteristics of the human ear. In the Mel scale, the frequency ranges are divided into four equal intervals. The frequency is adjusted there in such a way that one half of this frequency scale is equivalent to a given linear frequency. The Mel scale shows a good performance while discriminating the speech segments. In the Mel scaled band, the Mel scale is used to represent the frequency in the critical band. The filter bank is a set of triangular filter banks based on critical band scales. The spacing of the critical bands is non-linear. Next we discuss the relation between Mel scale and linear frequency scale measured in Hz. There are several different formulations of the Mel scale. Each of them is used differently in the literature. Now we present a definition of the Mel scale that is frequently used. With f the linear frequency in Hz is represented, and with $f_{mel}(f)$ is in the Mel scale formulated in Eq. 20.1. The frequency conversion from Hz to the *mel* scale uses the following formula:

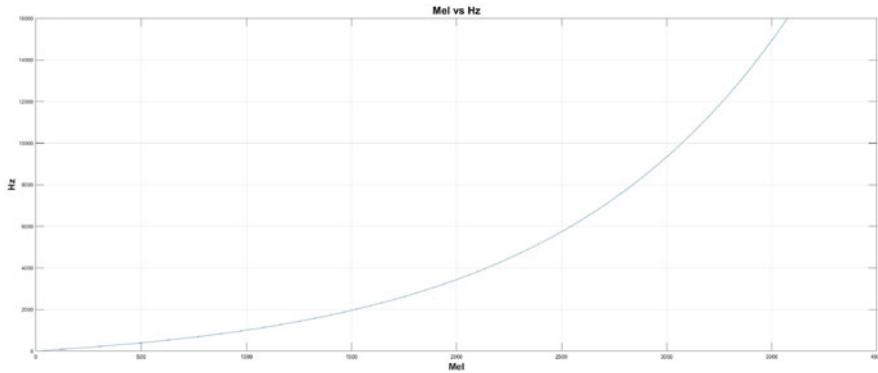


Fig. 20.2 Mel scale

$$mel(f) = 2595 \log_{10}(1 + \frac{f}{700}) \quad (20.1)$$

where, with f the frequency in linear scale is denoted, and with mel the frequency in Mel scale (Fig. 20.2).

20.3.2 Bark Scale

The Bark frequency scale is related to Mel scale in that has the property of an equal or uniform distance representing perceptually equal distances. The Bark scale is linear below 500 Hz and non-linear above 500 Hz. This non-linear spectral distance is measured using the logarithmic frequency axis. Thus the important attribute of the Bark scale is the width of the critical band at any given frequency, not the exact values of the edges or centers of any band. We state the formula for the Bark scale and its inverse transformation. Among the different Bark scale formulations:

$$\begin{aligned} Bark(f) &= 13 \arctan(0.00076f) + 3.5 \arctan((\frac{f}{7500})^2) \\ Bark(f) &= \frac{26.81f}{1960 + f} - 0.53 \end{aligned} \quad (20.2)$$

depicted in Eqs. 20.2, the formula shown in Eq. 20.3 provides the conversion from Hz to the Bark scale:

$$Bark(f) = 6 \sinh^{-1} \left(\frac{f}{600} \right) \quad (20.3)$$

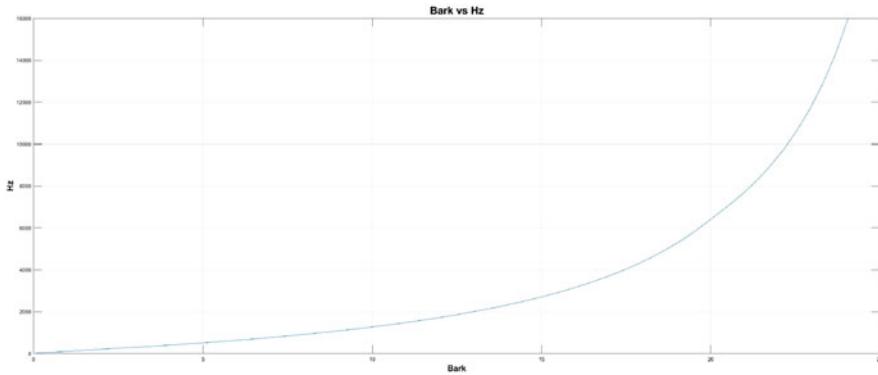


Fig. 20.3 Bark scale

where, with f the frequency in linear scale is denoted, and with *Bark* the frequency in Bark scale. The inversion from Bark to Hz is given by the following formula:

$$f = 600 \sinh \left(\frac{B(f)}{6} \right) \quad (20.4)$$

This relationship is depicted in the following Fig. 20.3 presented below:

20.3.3 Erb Scale

The other commonly used perceptual frequency scale is the Erb scale. The Erb scale is formulated in Eq. 20.5 where f is the center-frequency in Hz, normally in the range 100 Hz–10 kHz. The Erb scale is generally narrower than the classical critical bandwidth (CB) such as Bark or Mel scale and f is in Hz.

$$Erb(f) = 21.4 \log_{10} (0.00437f + 1) \quad (20.5)$$

The Erb warping is evaluated along a uniform frequency ranging from zero to the number of Erbs at half of the sampling rate, so that the direct current (DC) maps to zero and half of the sampling rate maps to π (Fig. 20.4).

20.3.4 Greenwood Scale

The Greenwood scale establishes a relation between the positions of the hair cells in the inner ear to the frequencies that stimulate their corresponding neurons. The function f in Eq. 20.6 makes this precise.

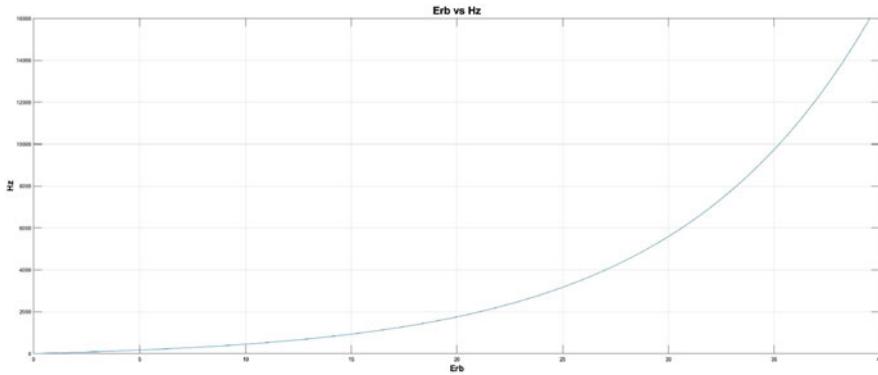


Fig. 20.4 Erb scale

$$f = A(10^{ax} - K) \quad (20.6)$$

where f is the characteristic frequency of the sound in hertz, A is a scaling constant between the characteristic frequency and the upper frequency limit, a is the slope of the straight-line portion of the frequency-position curve, which has shown to be conserved throughout all investigated measurements after scaling the length of the cochlea, x is the fractional length along the cochlear spiral measured from the apical end of the cochlea to the region of interest, where $0 < x < 1$, and K is a constant of integration that represents the divergence from the log nature of the curve and is determined by the lower frequency audible limit. A comparison between the frequencies of the introduced scales is shown in Fig. 20.5. In that figure, we see the comparison among the Bark, Erb, Greenwood and Mel perceptual scales. The curves indicating the scales show the similar behaviors against linear scale Hz. They are expanded at low frequencies below 1000 Hz and above this frequency they are all compressed.

20.4 Filter Banks

A filter bank is an array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal. The term filter bank is also used for a bank of receivers. The independent component analysis (ICA) network is successfully applied in separating of mixed speech sources. The basic ICA network consists of whitening, separation and basis vector estimation.

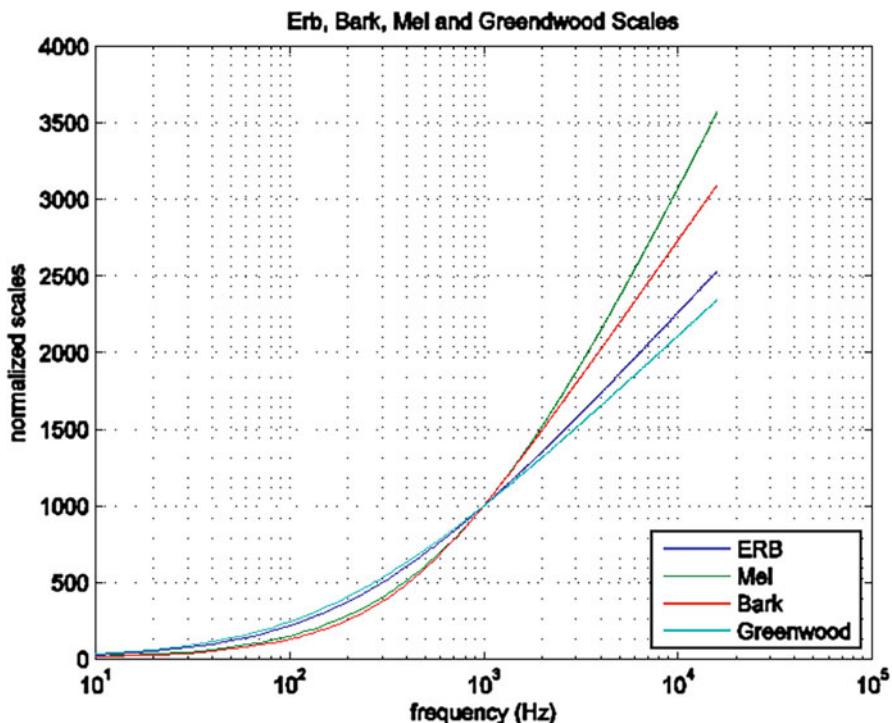


Fig. 20.5 Comparison of different scales

20.4.1 ICA Network

The ICA network approach is based on time-averaged audio spectral characteristics. However, this could not remove the detailed correlation which temporarily exists only at a time instance. Decimation of a filter bank is capable of removing both the averaged and detailed correlations. Among the filter bank approaches, oversampled filter banks, where the decimation factor is smaller than the number of analysis filters, accomplish better performance than critically sampled filter banks. The oversampled filter banks can have negligible aliasing when each filter has a high stop band attenuation, so they make it possible to perform adaptive filtering without requiring cross adaptive filters between adjacent bands or distorting reconstructed signals. Since ICA is performed in the oversampled filter bank, adaptive parameters in each sub band can be adjusted without any information from other sub bands. Thus, the filter bank approach is appropriate for parallel processing. The inputs, which are mixtures of unknown independent signals, are decomposed into sub band signals by analysis filters. Then, each sub band signal is down sampled by a decimation factor. Since the down sampled signals are still convolved mixtures whose reverberation length has decreased by the decimation factor, a typical ICA

algorithm for convolved mixtures can be used to obtain independent components from the down sampled signals at each sub band. Here, the unfixing filter length is much shorter than that of the full-band time domain approach. The outputs from the ICA network are expanded, and the original independent signals can be reconstructed from the sub band outputs through synthesis filters after fixing scaling and permutation. Usually, the length is shortened by a decimation factor, comparing with that of the corresponding adaptive filters in the full-band time domain approach.

20.4.2 *Auditory Filter Banks*

To capture the functions of the human auditory system and its hearing, a bank of filters is arranged in such a way that the pass bands of the filter banks are overlapped. This is used to model the human auditory system and one calls it the auditory filter bank. The filter shape can be triangular or trapezoidal. The filter bank can also be uniform or non-uniform. A set of transfer functions $H_m(z)$ in the analysis filter bank splits the input into M sub band signals. For the speech feature extraction usually non-uniform spaced filter-bank is used. In such a case, the part of the spectrum below 1 kHz is processed by more filters in the bank because it is assumed that the 1st formant lies in the lower frequency range and there exists more vocal tract information. The frequency resolution of the auditory filter bank largely determines which portions of a signal are perceptually irrelevant. The auditory time-frequency analysis that occurs in the critical band filter bank induces simultaneous and non-simultaneous masking phenomena that are assumed to be the distortion spectrum. A perceptual model exploits the masking thresholds for a complex sound. The loudness scale is related to the sound level depending on the duration and frequency of the sound. It is variable with respect to the perceived sound level.

20.4.3 *Filter Banks*

A filter-bank transforms an input signal into a sub-band domain and re-transforms back the signal into the input domain. The transformation from the time domain to the sub-band domain with M sub bands would entail an M -fold increase in the amount of data to be processed. In this case, the signal is first decimated into the sub band signals and such sub band can lead to aliasing. In such a case, the goal is not to alter the contents of the individual sub-bands but rather to digitally transmit or store them in such a way as to achieve the maximum possible fidelity with the minimum number of bits. This in turn requires the allocation of different numbers of bits to different sub bands. There are two types of filter bank architectures:

(i) **Perfect reconstruction:**

In this architecture the input signal to the filter bank and the output of the domain are the same. Any change or modification or distortion is not considered.

(ii) **Maximal decimation:**

In this structure the filter bank has M sub bands. The signals at the output of each of the M sub bands can be decimated by a factor M without losing the perfect reconstruction. This type property of transformation is utilized as the highest fidelity encoding with the fewest possible bits.

The filter bank can be uniform or non-uniform. A set of transfer functions $H_m(z)$ in the analysis filter bank splits the input into M sub band signals in the synthesis filter bank.

For speech feature extraction usually non uniform spaced filters are used. In such a case, the part of the spectrum below 1 kHz is processed by more filter banks because it is assumed that the 1st formant lies in the lower frequency range and there exists more vocal tract information. The filters are uniformly spaced below 1 kHz and then the filters are spaced according to logarithmic scale.

20.4.4 Mel Critical Filter Bank

The filters are combined in a filter bank denoted by $H_m(k)$ as defined in Eq. 20.7. As a reminder on the type of definition one calls them triangular filters. Suppose the number of filters is M and $m = 1, 2, 3, \dots, M$ where each m denotes a triangular filter. The frequency for the bin N is f . Here $n = 0, 1, 2, \dots, (N - 1)$. The filter bank $H_m(k)$ is given in Eq. 20.7.

$$H_m(k) = \begin{cases} 0 & \text{for } k < f(m-1) \text{ and for } k > (m+1) \\ \frac{2[k-f(m-1)]}{[f(m+1)-f(m-1)][f(m)-f(m-1)]} & \text{for } f(m-1) \leq k < f(m) \\ \frac{2[f(m+1)-k]}{[f(m+1)-f(m-1)][f(m+1)-f(m)]} & \text{for } f(m) \leq k \leq f(m+1) \end{cases} \quad (20.7)$$

The terms: $f(m - 1)$, $f(m)$, $f(m + 1)$ are the left, the middle and the right boundary. The boundary points are denoted by B and they are non-uniformly spaced in mel scale. Let f_l and f_h denote the lowest and the highest frequencies of the filter bank in Hz and f_s the sampling rate in Hz. Then one obtains relation depicted in 20.8:

$$f(m) = \frac{N}{f_s} B^{-1} \left[B(f_l) + m \frac{B(f_h) - B(f_l)}{M + 1} \right] \quad (20.8)$$

In this expression B denotes the acoustic scale, as mentioned earlier, it can be in mel, bark, or erb scale.

20.5 Psycho-Acoustic Phenomena

The psycho-acoustic quantities that are perceived can also be recorded. However, the recordings say nothing about their meaning. In this context we will describe the quantities that need to be extracted. This leads to a number of equations from psycho-acoustics area. In the inner ear, the basilar membrane is working as a spectrum analyzer. By responding to the temporal variation of the sound pressure wave and its localizations, the human ear responds to temporal variations of pressure and localizes the sound. The frequency, timing, amplitude, loudness and phase information at different frequency ranges and the localization of the sound sources are then analyzed by the brain. The speech signal in the temporal and spectral intervals (i.e. between 100 to 1000 ms) is generally analyzed by going through the audio sensation and its variations.

20.5.1 Perceptual Measurement

The perceptual measurement needs to be connected to the feature extraction. To adopt some basic psycho-acoustics quantities, we first review the human auditory system and its functioning and then the hearing model. The perceptual entropy is used for speech coding and multimedia application for speech data compression. The measurement follows a selection of some basic psycho-acoustics quantities such as frequency analysis and masking properties, perception of loudness and perceptual entropy. Some of them are commonly used in the perceptual feature extraction techniques. The behaviors of the Basilar membrane in the inner ear of the auditory system is similar to the overlapping pass-bands of a bank of band-pass filters. In the inner ear, the Basilar membrane is working as a spectrum analyzer. By responding to the temporal variation of the sound pressure wave and its localization, the human ear responds to temporal variation of pressure and localizes the sound. The frequency, timing, amplitude, loudness and phase information at different frequency ranges and the localization of the sound sources are determined by the brain. An adaptation to the critical bandwidth is related to the bandwidth of an auditory filter which is incorporated in masking, loudness, and absolute threshold.

20.5.2 Human Hearing and Perception

Hearing is one of basic signals that we perceive first. The processing time of the perceived sounds are following its temporal pressure variations. The characteristics of the sound perception process of the human ear and brain are non-linear. In the inner ear, the Basilar membrane is working as a spectrum analyzer. By responding to the temporal variation of the sound pressure wave and its localizations, the

human ear responds to temporal variation of pressure and localizes the sound. The frequency, timing, amplitude, loudness and phase information at different frequency ranges and the localization of the sound sources are determined by the brain. The study of the psychoacoustics is related to the perception of the sound and related phenomena. The speech signal in the temporal and spectral intervals i.e. between 100 to 1000 ms is generally analyzed by going through the audio sensation and its variations as well as the loudness-time function. The response of the human brain and ears can be quadratic, cubic or quartic. For example, two loud pure tones at corresponding frequencies f_1 and f_2 are simultaneously sounded together to generate a third difference tone $|f_2 - f_1|$ to be heard. The difference between a standard sub band analysis and a critical sub band analysis is that the standard sub band is of equal width and the width of the standard sub band does not reflect the human auditory behavior where in the critical band analysis it works according to the function of frequency to approximate the human auditory behavior. The critical band based sub band uses some scales to follow the distance of the Basilar membrane in the cochlea in the inner ear. The critical band analysis is mapped to the critical band frequency scale such as Mel, Bark or Erb scale. In order to model the human auditory system the perceptual quantities such as absolute threshold of hearing (ATH), sound pressure level (SPL), sensation level (SL), masking frequency, temporal masking, the mapping of the non-linear frequency scale are considered. In the frequency analysis the signal is transformed to a non-uniform logarithmic scale following some special frequency scale (such as Bark, Mel or ERB). The mapping process and the transformed non-uniform new scale is called a critical band. In psychoacoustics the values of the sound pressure lies between 10^{-5} and 10^2 Pascals. It is measured by a hearing threshold given by SPL. Then again in the perception stage, there is a sensation level (SL) which indicates an intensity level of acoustic events to be heard by a listener. The SL may be used sometimes to determine which sound to be heard regardless of the loudness of the sound. It is not the same as ATH. In principle, the techniques developed for psychoacoustic phenomena apply to other than speech signal processes. This is important where one has not a precise semantics shared both by sender and receiver. This is for instance the case for applications in medicine and in the seismic area. A typical element of psychoacoustics is loudness. This cannot be expressed in written speech. One way to measure a quantity related to it is the temporal cumulative loudness distribution. The distribution gives the percentage of time of a given loudness level. This provides a scale and the user can indicate what this means when hearing to it. When doing so a comparison between the scale and the scale of the loudness perception of the human ear is needed. A concept for this is the sound pressure level discussed next.

20.5.3 Sound Pressure Level (SPL)

The ratio between the reference sound pressure in Pascals and the threshold of hearing in Pascals is the sound pressure level (SPL). The SPL is measured in dB.

It plays a role for human hearing. The absolute threshold of hearing is estimated as $p_0 = 2 \times 10^{-5}$ Pascals which is about $20 \mu Pa$. The intensity of the sound pressure in decibels relative to a given reference level is computed by Eq. 20.9. In this equation, L_{spl} is the sound pressure level, p is sound pressure of an event in Pascals.

$$L_{spl} = 20 \log_{10} \frac{p}{p_0} \quad (20.9)$$

The relation between dB and Pascal is shown below in Eq. 20.10, and the dB to Pascal transformation is shown in the next Eq. 20.11 where Pa stands for Pascal. The dB_{SPL} means SPL is measured in dB .

$$L_p(dB_{spl}) = 20 \log_{10} \frac{p}{p_0} \quad (20.10)$$

$$p(Pa) = p_0^{\frac{L_p(dB_{spl})}{20}} \quad (20.11)$$

The SPL is also measured with respect to sound intensity level (SIL) in dB . This is equal to the sound power level (PWL). Auditory masking is a psychoacoustic effect that determines the mapping of the frequency in the critical band. Frequency masking makes one sound inaudible due to a presence of another sound. This gives the threshold of the audibility at where one sound is raised by the presence of another sound. The frequency action, SPL is the sound pressure level, p is sound pressure of an event in Pascal.

20.5.4 Absolute Threshold of Hearing (ATH)

The threshold of hearing denotes a listener's ability to recognize a sound in a noise free environment. This is expressed by a sound pressure level (SPL) and computed by Eq. 20.12 where f is the frequency in hertz and $T_q(f)$ is expressed in dB .

$$T_q(f) = 3.64 \left(\frac{f}{1000} \right)^{-0.8} - 6.5e^{-0.6 \left(\frac{f}{1000} - 3.3 \right)^2} + 10^{-3} \left(\frac{f}{1000} \right)^4 \quad (20.12)$$

The threshold of hearing is measured in SPL in dB in a quiet environment as a function in SPL in dB . This is a standard formula used in psychoacoustics studies. The threshold of hearing is shown in Fig. 20.6.

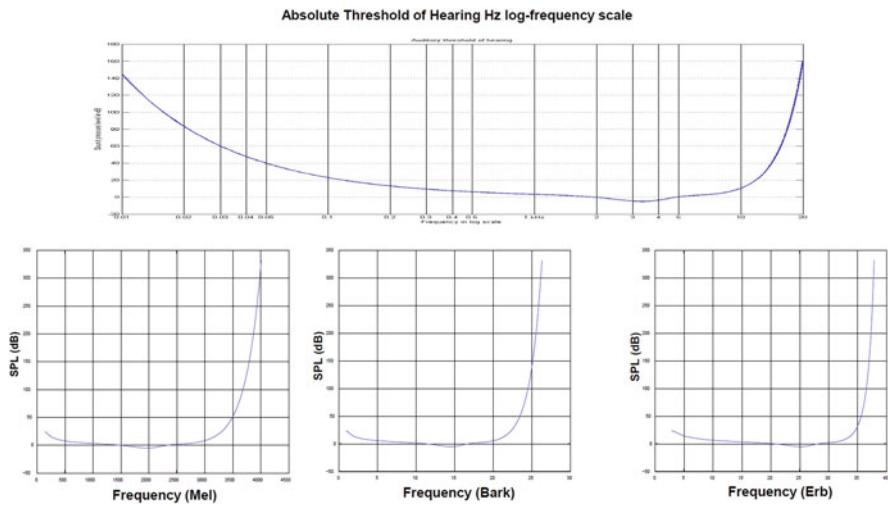


Fig. 20.6 Threshold of hearing

20.6 Perceptual Adaptation

The purpose of a perceptual model is to hear, interpret and understand the sounds (e.g., of spoken language). We concentrate on the first and partially on the second issue. The speech sound signal contains a number of acoustic elements that are used in the speech perception. These representations can then be combined to be used in the word recognition and other language processing activities. This is done particularly in the cochlea and in the basilar membrane. The perceptual adaptation is managed by adopting several perceptual quantities such as the critical bandwidth transformation, the intensity-loudness power law transformation which is also a kind of hearing law. Next we introduce the hearing process of the human ear that we experience in our daily life in order to recognize speech.

20.7 Auditory System and Hearing Model

The sound enters into the human ears as a pressure wave and the human ears perceive the sound by its vibration. The human ears are also known as an auditory system because this acts as a sensor for the human hearing. For this, the human ears are the principle organ. The understanding of this organ and its activities can significantly contribute to building automated systems that are used for speech recognition. We remark again that understanding refers to the syntax and not to semantics.

Here we first explain how the human ear interprets the sounds for its perception. Then we introduce how the human ear is used in the literature in order to model it.

20.8 Auditory Masking and Masking Frequency

In ordinary life masking means to make something invisible. This applies to hearing too. Auditory masking is a psychoacoustic effect that determines the mapping of the frequency in the critical band. Frequency masking makes one sound inaudible due to a presence of another sound. This gives the threshold of the audibility where one sound is raised by the presence of another sound. This relates the frequencies and they way humans perceive it. The inaudible frequency of the sound one calls this a masked frequency and the frequency of the sound which presence makes masked frequency is called masker frequency. Two common masking types are:

- Frequency masking or simultaneous masking excites multiple tones at the same time and
- Temporal masking excites a particular frequency zone in the cochlea along the Basilar membrane.

One carries over both types of masking to the human brain by the auditory nerve. The auditory masking is related to the standard sound pressure level (SPL) and the sensation level (SL) which is an intensity level of an acoustic event to be heard by a listener. The masking effect is the same when the power of the tone and the power of noise spectrum is near that tone. However, the masking effect outside this area of the tone does not interfere to that described area. Here, if the characteristic frequency band has the same acoustic power for the tone and the noise spectrum within that band, the tone is masked. Further to this explanation, an assumption is that the human hearing system processes sounds in relatively narrow frequency bands. The hearing system produces masked threshold frequencies independent of the frequency. The unmasked threshold is the quietest level of the signal which can be perceived without masking the signal. In some literature, the total masking threshold is approximated by a summation of the threshold produced by individual signal components following the power law.

Simultaneous masking occurs when a noise or unwanted sound of the same duration as the original sound makes a sound inaudible. For example, a powerful spike at 1 kHz will tend to mask out a lower-level tone at 1.1 kHz. Also, two sine tones at 440 and 450 Hz can be perceived clearly when separated. They cannot be perceived clearly when presented simultaneously.

Temporal masking or non-simultaneous masking occurs when a sudden stimulus sound makes inaudible other sounds which are present immediately preceding or following the stimulus. Masking which obscures a sound immediately preceding the masker is called backward masking or pre-masking. Masking which obscures a sound immediately following the masker one calls forward masking or post-masking. Temporal masking's effectiveness attenuates exponentially from the onset

and offset of the masker, with the onset attenuation lasting approximately 20 ms and the offset attenuation lasting approximately 100 ms.

The human sound perception i.e. the speech hearing is affected by masking properties. The intensity the acoustic stimulation is measured by the standard sound pressure level (SPL). The loudness remains constant for a narrow band noise source at a constant SPL even as the noise bandwidth is increased up to the critical bandwidth tends to remain constant about 100–500 Hz audio increases approximately 20% of the center frequency above 500 Hz.

The width of the critical band is commonly referred to as one Bark scale which is a non-linear function. It is often used to convert the frequency from the Hertz to the Bark scale. In order to model the human auditory system, the perceptual quantities such as absolute threshold of hearing (ATH), sound pressure level (SPL), sensation level (SL), masking frequency, temporal masking, the mapping of the non-linear frequency scale are considered. The ratio between the reference sound pressure in Pascals and the threshold of hearing in Pascals is formulated in Eq. 20.12.

20.9 Perceptual Spectral Features

To deal with ordinary speech is somehow difficult. The spectrally shaped speech can now be mapped to psychoacoustic quantities in order to approximate the human speech perception. The psychoacoustic quantities are:

- Intensity loudness.
- Critical band analysis,
- Equal loudness pre-emphasis

These will now be discussed.

20.10 Critical Band Analysis

In order to model the human hearing we use the Bark scale. The formulation of the Bark critical band analysis is given earlier in this chapter. In application of filters, the process of windowing is explained in Chap. 3 in Sect. 3.1.8.1.

Each of the filter's output is the sum of the product of the windowed FFT speech signals and the $\Phi_m(k)$ weight. There m is the running index of the filters in the filter bank and k denotes the frequency of certain m th filter. In 20.7 the first filter (at $m = 1$) is at 0 Barks and the last filter is at $m = M$. The outputs of the first and last filters are calculated in such a way that the resultant output of these two filters is equal to their closest filter output.

$$X_m(k) = \sum_{k=0}^{\frac{N}{2}-1} |s(k)|^2 |\Phi_m(k)| \quad (20.13)$$

where with Φ_m we denote the filter output of the m 'th filter and $|S(k)|^2$ is the N th power of the windowed speech frame, and $|\Phi_m(k)|$ is the filter weight of the m 'th bark filter corresponding to the linear frequency scale.

20.11 Equal Loudness Pre-emphasis

This re-emphasizes the spectrum to approximate the unequal sensitivity of human hearing versus linear frequency to approximate the equal loudness curve. This is known as the psychophysical equal loudness pre-emphasis which reduces the spectral amplitude variation between the critical band spectrum and the linear frequency band spectrum. Up to now we set loudness up several times as a topic. Unfortunately there is no formal concept that covers all intuitive interpretations of the term. The following equation is sufficient for all interpretations used in this book and can serve as a basic for various implementations. E_m is the weighted loudness.

$$\eta_m(k) = \sum_{k=0}^{\frac{N}{2}} E_m(k) X_m(k) \quad (20.14)$$

In Eq. 20.15, with ω the angular frequency is represented where $\omega = 2\pi f$, m denotes the signal segment and f is the linear frequency. Thus

$$E_m(\omega) = \frac{(\omega^2 + 56.8 \times 10^6)\omega^4}{(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)} \quad (20.15)$$

Given the signal intensity, η approximates the human loudness perception. η is a cubic-root amplitude compression which relates the sound intensity and perceived loudness.

$$\zeta_m(k) = \sum_{k=0}^{\frac{N}{2}} \left(\eta_m(k) \right)^{\frac{1}{3}} \quad (20.16)$$

20.12 Perceptual Transformation

Perceptual transformation uses the cepstral coefficients of the spectral speech features. It uses the inverse discrete transform of the perceptually weighted spectral speech vector for its feature transformation. The inverse of features makes their value for humans visible. It is illustrated in Eq. 20.16. This inverse relation and the dependencies of the inverse features is shown in Eq. 20.17.

$$\zeta_m(k) = \frac{\sum_{n=0}^{N-1} S_m(k) e^{j \frac{2\pi}{N} kn}}{N} \quad (20.17)$$

20.13 Feature Transformation

Here we use the autoregressive approach for the perceptually transformed features for a vocal tract modeling. It uses also the solution to the parameters obtained by using LPC analysis. It contains the components that have been already introduced in this technique. More about the intensity and loudness can be found in psychoacoustics studies. We have not done a detailed investigation of these studies. These features in each frame are good managed but it shows some constant value. This might be the energy extracted from each feature vector.

20.14 Filters and Human Ear

The spectral analysis and parametric representation attempts to approximate the spectrum analysis of the human auditory system. There are different representations in the ear what is quite surprising. Comparing acoustic scales such as Mel scale or Bark-scale in the filter bank is commonly done. The speech sounds goes into the outer and middle ear and then into the inner ear where the Basilar membrane is seen as an spectrum analyzer of the input sound and the hair cells convert the signals into neural signals as depicted in Fig. 20.7. The auditory nerve carries the neural signal to the brain for its processing. This transfer is presently not fully understood because the knowledge about the functioning and the structure of the brain. The topic of the brain is not discussed here.

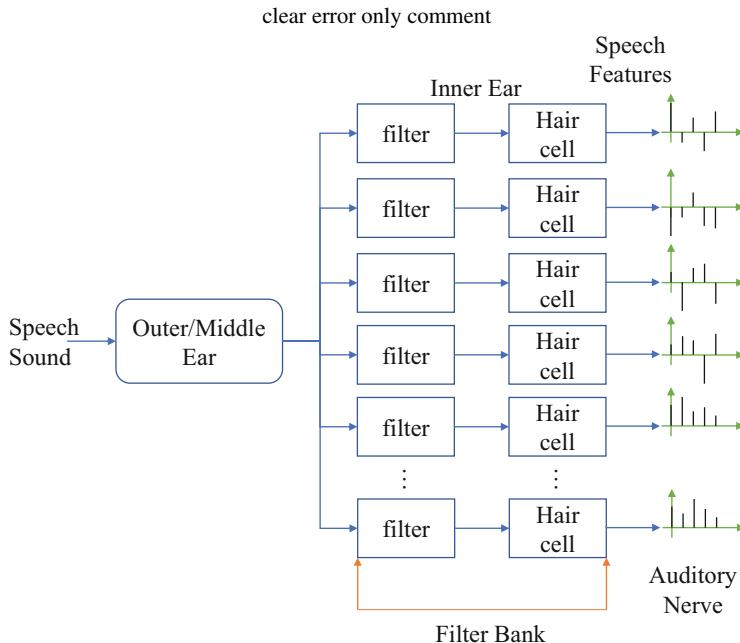


Fig. 20.7 Filters in the human Ear

20.15 Temporal Aspects

One important aspect why speech recognition differs from reading is that in a written text no temporal information is contained. It is just a sequence of words or possibly illustrations. On the other hand spoken words are given in time and the way it is given can contain information to the listener. Hence a single word says very little. The kinds of temporal distances between words also contain information. For instance a silence in a spoken sentence may express some criticism. We discuss this in the section on psychoacoustic phenomena. However, we are considering just the syntactic elements and not the meaning, i.e. the semantics. However, temporal aspects play a significant role in other applications of signals. Hence they should be regarded as an aspect of the signal process. Important examples are medical and seismic applications that will be discussed below. There the role of silence intervals plays a significant role. According to cultural norms, silence in speech can be positive or negative, for instance in churches. If the silence in heartbeat is long this may be indicative of heart decease. This is important for medical applications discussed below.

20.16 Background Information

General

We took the picture of the human auditory system from Zanker (2013). It shows all parts of the body participating in the recognition. The names of the scales have different origins. The name “mel” comes from the word “melody”. There is no uniform measure for mel scale but there are several proposals.

Past

The name bark is short for Heinrich Barkhausen who proposed the first subjective measurements of loudness. More on the bark scale is in Ellis (2005). As mentioned in section 11.5 they all give the same information about the relation between the Hz and the Bark scale. ERB is an abbreviation of “Equivalent Rectangular Bandwidth”. This is more detailed in Smith (2011). Greenwood scale is named after Donald D. Greenwood who introduced it in 1961. Nolan (2003) is concerned with scales. Filters have an input and output and they perform certain transformation. Filters are used in many aspects of audiology and psychoacoustics including the peripheral auditory system. Auditory filters are closely associated with masking in the way they are measured and also the way they work in the auditory system. See also Glasberg et al. (1990). ICA network is described in Juha Karhuhnen, see Karhunen et al. (1997).

Suggested

A Bark-scale filter bank approach to independent component analysis for acoustic mixtures is available from <https://www.researchgate.net/publication/223801336>. Speech and audio techniques are discussed in Gold and Morgan (2000). For more on psychoacoustics one can look at Kahrs and Brandenburg (2002), Johansen (2006) and Dutoit and Marques (2009). Johansen gives a threshold for the human hearing. This is also used for discovering properties of signal processes many applications where speech is not involved. 2002. Psychoacoustics is also the scientific study of sound perception. For more information about perception consider Davis and Mermelstein (1980). More for using spectral analysis is in Paiter and Spanias (2000) and Moore (1995). An early reference about sound pressure level is Longhurst (1967) and a more modern one iThe perceptual adaptation goes back to Hermann Helmholtz around 1800. A modern view is in Myers (2007). Masking is discussed in Moore (1986a,b). More about the intensity and loudness can be found in the psychoacoustics studies of Fast and Zwicker (2007). Neuroanatomical differences in brain areas implicated the perceptual and other core features of autism are revealed by cortical thickness analysis and voxel-based morphometry. Morphometry is the characterization of form of objects by quantitative measurement numbers. This plays a role if images are involved.

Multiple references such as Fletcher (1933, 1940), NIH (2017); <http://books/w3k.org>; Cheng et al. (2005), Landman (2001), Wolfe (2014), and Huang et al. (2003) are used in multiple occasion in this chapter. Readers are encouraged to take a look at these references for more details.

20.17 Exercises

Exercise 1 Give example of a sentence spoken by a human where different distances between the words give different impressions to the listener.

Exercise 2 Do the same as in exercise 1 for loudness.

Exercise 3 Suppose you want to model that some signal sequence expresses a question. In which kind of psychoacoustic elements would you be interested?

Exercise 4 Do the same as in exercise 4 for a signal sequence expressing strong opposition.

References

- [Fletcher1933] Fletcher, Harvey (1933). “Loudness, its Definition, Measurement and Calculation”, <https://archive.org/details/bstj12-4-377>, Bell System Technical Journal, October 1933.
- [Fletcher1940] Fletcher, Harvey (1940). “Auditory Patterns”. Reviews of Modern Physics. 12 (1): 47–65.
- [NIH2017] Journey_of_Sound_to_the_Brain.ogv”, National Institute on Deafness and Other Communication Disorders, part of the National Institutes of Health, 20 December 2017
- [Zanker2013] Zanker, J.M. (2013). Auditory perception: Hearing noise and sound. PS 1061.
- [Davis1980] Davis, S.B., Mermelstein, P. (1980): Comparison on parametric representations for monosyllabic word recognition in continuously spoken sentences. Acoustic, Speech and Signal Processing 28.
- [Smith2011] Smith J.O. (2011). Spectral Audio Signal Processing. W3K Publishing [w3k] <http://books.w3k.org/>.
- [Ellis2005] Ellis D. P. W. (2005): PLP and Rasta, and mfcc, and inversion. In matlab Wavepage.
- [Chengetal2005] Cheng, O. Abdulla, W. Salcic, Z. (2005): Performance evaluation of front-end processing for speech recognition systems. School of Engineering Report 621, Faculty of Engineering, University of Auckland.
- [Johansen2006] Johansen, L.G. (2006). Psychoacoustics and audibility -fundamental aspects of the human hearing. Lecture notes TI-EAKU, University College of Aarhus, 2006.
- [Landman2001] Landman D.R. (2001). Design of a sound level meter. Laboratory Report EN 253: Matlab Exercise, November 200 5.
- [Longhurst1967] Longhurst, R.S. (1967). Geometrical and Physical Optics. Norwich: Longmans.
- [Wolfe2014] Wolfe, J. (2014) What is acoustic impedance and why is it important? University of New South Wales, Dept. of Physics, Music Acoustics. 2014.
- [Huang2003] Huang X, Acero A., and Hon H.-W. (2001). Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall.
- [Nol2003] Nolan F. (2003). Intonational equivalence: An experimental evaluation of pitch scales. Number 15. International Congress of Phonetic Sciences.
- [Fast2007] Fast H., Zwicker E. (2007). Psychoacoustics: Facts and Models. Springer, Deutschland, Heidelberg.
- [Kahrs2002] Kahrs M., and Brandenburg K. (2002). Applications of Digital Signal Processing to Audio and Acoustics. Kluwer Academic Publisher, Boson, USA
- [Myers2007] Myers, David G (2007) Exploring Psychology in Modules (7th ed), New York, Worth Publishers
- [Glasberg1990] Glasberg, B. R., Moore, B. C. J., (1990). Derivation of auditory filter shapes from notched-noise data.

- [Dutoit2009] Dutoit T., Marques F. (2009). Applied Signal Processing. Springer Verlag.
- [Moore21986] Moore, B. C. J., (1986a), Parallels between frequency selectivity measured psychophysically and in cochlear mechanics. Scand Audio Suppl 1986.
- [Painter2000] Painter, T Spanias, A. (2000). Perceptual Coding of Digital. Proceedings of the IEEE Audio, 88.
- [Gold2000] Gold, B., Morgan, N. (2000). Speech and Audio Signal Processing. John Wiley and Sons Inc, New York.
- [Moore21986] Moore, B.C.J. (1986b) Frequency Selectivity in Hearing, London, Academic Press.
- [Moore31995] Moore, B.C.J. (1995) Handbook of Perception and Cognition Press, Burlington, MA, USA, 2nd edition, 1995.
- [Karhunen1997] Karhunen, A., Oja, A., Wang, L., Vigario, R., Joutseno, J., A class of neural networks for independent component analysis, Vol 3, No, 3, IEEE Transactions on Neural Networks, 1997

Chapter 21

Speech Features



Overview

Feature extraction is an important stage in pattern recognition. For algorithmic solutions, a mathematical model represents knowledge about the problem which is prior knowledge defined in input and output space. Typical model is search based model which has been well formulated in machine learning. Such standard learning algorithms are classification, clustering and parsing. The classification learn the parameters for function from model inputs to classes, clustering learn search algorithm parameters for detecting regions of interest, parsing learn search algorithm parameters for structural descriptions such as trees, graphs, etc. In speech recognition, the speech waveform as a physical signal converted to digital signals, vector patterns are constructed from feature extraction, then classification is used in the post processing for making decision about the speech. In this chapter, we will see the typical speech features, how they are processed using different perceptual scale, and how the extracted speech features are constructed for decision making.

21.1 Generalities

In this chapter we extend the concepts of the Part I and II in many respects in the direction of problems coming from applications. A very important application is speech recognition. There one has several special aspects that have to be considered. For other applications, as in the bio-medical area, its importance can not be overstated. Features for general signal sequences and their extraction have been discussed in Chap. 11. They play a central and special role for speech recognition. The short-term spectrum is commonly used in all speech feature extraction methods. The LPC is used as all-pole auto-regressive (AR) model in the decomposition of speech production model. The LPC extracts the set of prediction coefficients from the speech frame (5–30 ms in duration) of the speech signal. The MFCC is based on

the filtering of spectrum using the mechanism of the human speech perception. The LPCC is based on the autocorrelation of the speech frame. The use of MFCC in the front-end results in better recognition accuracy than LPCC. However, the LPCC requires less computational steps than MFCC, etc. One has different tasks that depend on the information one wants to keep after the extraction. For instance, with respect to psychoacoustics phenomena one wants information about the distance between the words. The human ear has techniques incorporated for dealing with this problem. The task now is to enable a machine to perform it.

A feature vector contains a finite number n of real numbers. The idea is now to transform the large number of possible representations into a very small number of feature vectors that carries all essential information about the speech. One calls this transformation feature extraction and was indicated in Part II. One distinguishes between two types of extractions:

(i) **Parametric Fourier Transform** based speech feature extraction:

This commonly uses spectral envelope of the speech in the transformation for feature extraction.

(ii) **Non-parametric Wavelet or Local Trigonometric Transform (LTT)** type of feature extraction:

These extraction processes may be categorized as non-parametric because they do not use a model. They rather decompose the signal in a special manner and feature analysis is generally based on a discrete cosine transformation.

We will now continue further to discuss LTT. The LTT has a particular advantage and it is similar to the Wavelet Packet Transform in that regard. This method enables a flexible adaptation of the sizes of the time and frequency windows. In sub-intervals where the frequency spectrum remains rather constant in time (as it is the case for vowels), the time window is adapted to a large size (low temporal resolution). In order to extract the occurring frequencies as precisely as possible (high frequency resolution) we need to apply smaller size windows. The opposite is the case where a fine subdivision of the temporal interval is performed to focus on temporal changes of the spectrum rather than on the precise values of the frequencies. Regardless of the difference in the feature extraction methods, spectral shaping and spectral analysis are common in extracting speech features.

21.2 Cost Functions

Feature extraction is as an activity related to costs. These costs are not only of financial character. They are incurred and are represented in terms of times (or complexities) of the performance. There are several possibilities to compute the cost function. Examples are Shannon entropy, Neumann entropy, wavelet entropy etc. The Shannon spectral entropy is a common such cost function. If a set of signals has a maximum entropy probability distribution function (pdf) then this implies that the signals are mutually independent. However, a set of independent signals

does not necessarily have a pdf with a maximum entropy. According to Shannon's explanation some random processes such as speech have an irreducible complexity below which the signal cannot be compressed further and this limit specifies its entropy. Entropy, relative entropy, mutual information are some basic terms in probability distributions. Some feature extraction techniques incorporate perceptual properties of human hearing and human speech production. The requirements and the representations can be different for different applications. One cannot say in general what the best feature extraction method is. Such representations vary according to the type of the demand. Each demand prefers a specific feature extraction method.

21.3 Special Feature Extractions

Below we provide some different measurements that are commonly used as features in the speech recognition. With respect to the feature extraction stage in the speech recognition studies one considers mostly MFCC, LPCC, and PLPCC. In MFCC and PLPCC the signal decomposition and spectral analysis are followed by the process of the lapped transformation where the FFT is applied. The problem of abrupt discontinuity is present although its effect is reduced because of the lapped transformation. There exist non-standard LTT approaches. But the feature extractions in LTT do not make use of all available information provided by the speech. Another problem in the LTT transformation is that it is not transformed to a perceptual conversion.

The LTT has been used for speech processing, seismic data processing, speech recognition. There a perceptual mapping is not used, while it is used in speech processing and speech recognition. The most popular one is MFCC, in particular in the context of speech signals.

21.3.1 MFCC Features

The MFCC is based on human perception of pitch, which maps frequency of the signal non-linearly. Davis and Mermelstein introduced MFCC in 1980. It is the most commonly used feature extraction method in speech recognition. MFCC has become standard in solving speech recognition problems since 1980s. It is a combination of non-uniformly spaced filters with the combination of inverse discrete Fourier transform (IDFT). It is derived from a cosine basis expansion of the log spectral energy. MFCC uses several steps such as cepstral analysis, filter banks, IDFT for feature extraction. In such an algorithm, the signal spectrum passes through a triangular filter bank spaced on a linear-log frequency scale, and the energy output from each filter is compressed and transformed via the Discrete Cosine Transform (DCT) to cepstral coefficients. The MFCC tries to approximate

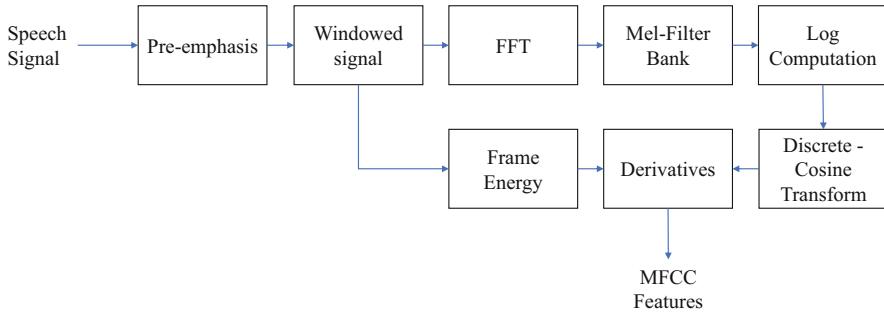


Fig. 21.1 MFCC feature extraction

the human hearing. It extracts perceptual speech features to be used for the speech recognition. It is probably the first perceptual speech feature extraction technique for the speech recognition system. This is one of the most commonly used feature extraction techniques in the automatic speech recognition (ASR). Figure 21.1 gives an overview of the MFCC feature extraction technique and shows how it is obtained.

Before the computation starts we have the pre-emphasis as an introductory step.

21.3.1.1 Pre-emphasis

The time domain representation of the pre-emphasis filter of an input signal x is denoted by s_m . Here the pre-emphasized output signal s_m at time n is a difference of input sample $x[n]$ and its delayed version of the original $x[n - 1]$ as formulated by Eq. 21.1

$$s_m[n] = x[n] - \alpha x[n - 1]; \quad (21.1)$$

where input is x , and output is denoted by s_m . Example of pre-emphasised signal is provided in Fig. 21.2.

We discussed the windowing in detail in Chap. 11 in Part II. In this context it is applied here. In the sequel it is assumed that windowing has been done. But first several other concepts have to be considered. Each speech signal is divided into several frames; each frame covering 5–30 ms of speech which ensures quasi-stationarity of speech. Each frame of signal corresponds to a spectrum (realized by FFT transform) (Fig. 21.3).

The spectrum represents the relationship between frequency and signal energy. Mel filter refers to a number of band-pass filters. In Mel frequency, the passband of the band-pass filter is the same width, but in Hertz spectrum, Mel filter has narrow dense cut-off band at low frequency, sparse high frequency and wide passband, aiming to simulate the perception of non-linear human ear to sound by having more discrimination at lower frequency and less discrimination at higher frequency. This

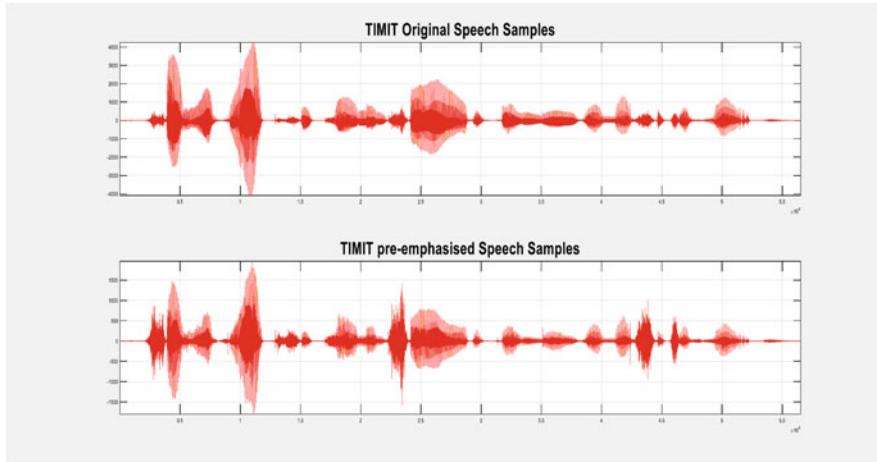


Fig. 21.2 Example of pre-emphasised data computed from TIMIT corpus: TIMIT/TRAIN/DR1/FSJK1/SI696.WAV

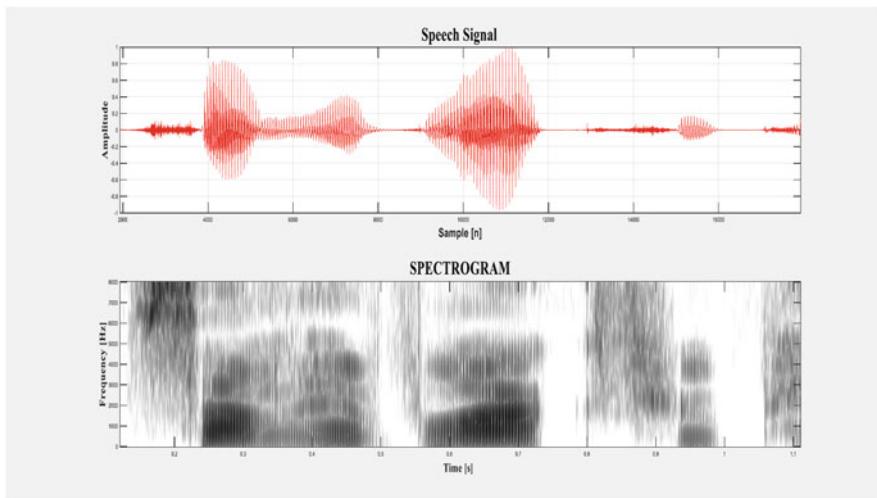


Fig. 21.3 Example of FFT data computed from TIMIT corpus: TIMIT/TRAIN/DR1/FSJK1/SI696.WAV

is further discussed in Chap. 24. The relationship between Hertz frequency and Mel frequency scale is given in Chap. 20 Sect. 20.3.1 and corresponding triangular filter bank is provided below (Fig. 21.4):

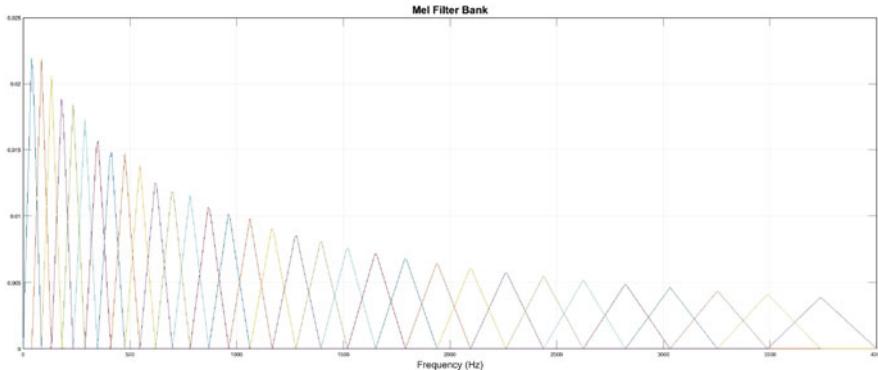


Fig. 21.4 Triangles of different bandwidths following Mel scale

21.3.1.2 Logarithmic Power Spectrum

In this step, one computes first the logarithm of the square of the spectrum of the windowed signal $s_m[k]$. One does this by taking the logarithmic computation of the square of the absolute value of the magnitude of the DFT for the windowed signal. This is shown in Eq. 21.2

$$s_m[k] = 10 \log_{10} \left| \sum_{n=0}^{N-1} s_m[n] e^{-j \frac{\pi}{N} kn} \right|^2 \quad (21.2)$$

21.3.1.3 Perceptual Spectral Analysis

The human ear has a high frequency resolution in the low frequencies and a low frequency resolution in the high frequencies. In order to reflect the frequency resolution property of the human ear, the logarithmic power spectrum shown in Eq. 21.2 is multiplied by the Mel scaled triangular filter banks. The Mel scaled filter bank is known as Mel filter bank. This is a set of triangular shaped filter banks which are scaled by a Mel scale. The number of filters is M and $m = 1, 2, 3, \dots, M$, where m denotes the triangular filters and f is the frequency in each bin whose size is equal to the FFT size. Now the filter-bank $H_m(k)$ is given in equation below. The $f(m-1)$, $f(m)$, $f(m+1)$ are the left, middle and right boundary of the m th filter. $H_m(k)$ is the weight of energy at frequency k for the m th filter.

21.3.1.4 Logarithmic Perceptual Power Spectrum

In this step, the logarithm of the squared magnitude of the output of the Mel filter bank is computed. The logarithm of the spectrum of each filter-bank is obtained on each frame using Eq. 21.3. The reason for computing the logarithm on the Mel power spectrum of the speech frames is to compress the wide-ranging varieties of input speech to follow the dynamic range compression characteristics of the human hearing system. In Eq. 21.3 the logarithmic power spectrum of each filter is $P_m(l)$ for $l = 1, 2, \dots, L$ many triangular band pass filters and $m = 1, 2, \dots, M$.

$$P_m[l] = 10 \log_{10} \left| \sum_{n=0}^{N-1} s_m[n] e^{-j \frac{\pi}{N} kn} \right|^2 \quad (21.3)$$

21.3.2 Feature Transformation Applying DCT

Since the log power spectrum is real and symmetric, the inverse DFT is equivalent to the discrete cosine transformation (DCT). The DCT produces uncorrelated features and thus the feature variations can be set using a diagonal assumption. This property is used to model the speech features using the Gaussian mixture model (GMM) for the classification in the HSM recognition. But this type of recognition is not considered here. We apply the MFCC features for pattern matching using DTW. The DCT of $P_m(l)$ is computed using equation depicted in Chap. 11 where $l = 1, 2, \dots, L$ and $n = 0, 1, 2, \dots, N$. The notations () and [] are used to denote the time and frequency domain index.

21.3.2.1 PLP

PLP stands for “Perceptual Linear Prediction” and it is a variant of the cepstral analysis. It relates the psychoacoustics studies to the auditory spectrum in order to create perceptual PLP features (Fig. 21.5).

The LP analysis uses the cepstrum analysis and the insights about human perception. In PLP analysis, initially, the spectral envelop of the spectrum of the windowed signal is measured. Here the concept of scales is used. It is discussed in

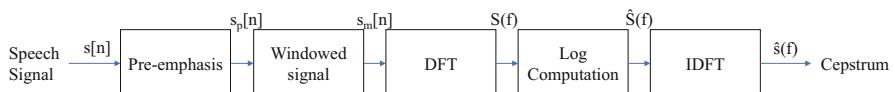


Fig. 21.5 Block diagram of cepstrum features generation

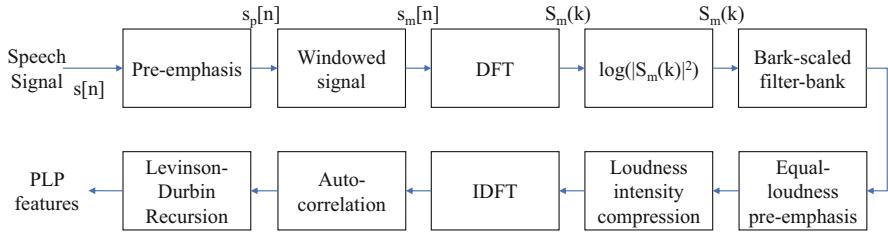


Fig. 21.6 Block diagram of PLP features generation

detail in Sect. 20.3. For perceptual analysis a Bark-scaled non-linear frequency scale is specifically used. The relation between the Bark scale and the linear frequency scale is given in Eq. 21.4. There f is the frequency in Hz and f_b is the related Bark frequency in Bark scale.

$$f_b = 6 \ln \left(\frac{f}{600} + \left(\left(\frac{f}{600} \right)^2 + 1 \right)^{0.5} \right) \quad (21.4)$$

The center frequencies of the filters in the filter bank is spaced in Bark scale and the center frequencies of these filters are approximately apart from each other by 1 in Bark scale. The first filter is at zero frequencies and the last filter is at Nyquist frequency. Therefore, the lowest frequency in the filter bank is 0 Nyquist frequency which is similar to Mel scaled filter bank. A more detail steps are shown in Fig. 21.6 are:

- **Spectral Shaping:** This is the same as in the cepstrum analysis as described in MFCC analysis.
- **Perceptual Spectral Analysis:** It uses the human speech perception. With critical band analysis, equal loudness pre-emphasis and intensity loudness conversion and adapt them to the speech which are already spectrally shaped.

In MFCC and PLP the signal decomposition and spectral analysis is followed by the process of the lapped transformation where the FFT is applied. The problem of abrupt discontinuity is due to fixed analysis window length.

- **Perceptual Feature Transformation:** This is the same as in the cepstrum analysis as described in MFCC analysis.
- **Feature Transformation:** It uses the human vocal tract model of perceptual feature transformed speech and then solves the problem of finding the model parameters using LPC analysis. The resulting parameters are called the PLP features.

The block diagram shows the complexity of PLP structure where the fundamental steps used in MFCC and PLPCC are depicted as follows:

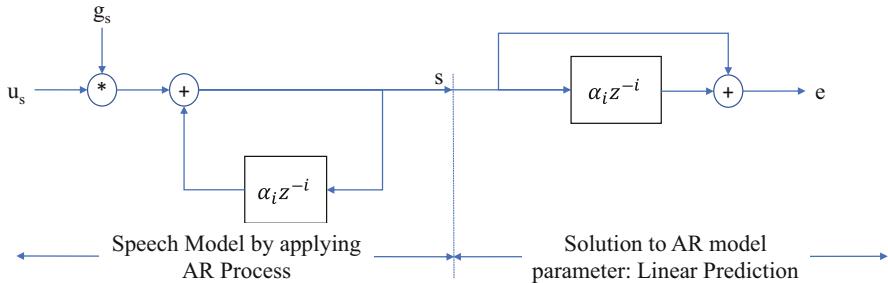


Fig. 21.7 Linear Prediction following the source excitation model

- Segmentation by Windowing
- Power Spectrum Computation
- Perceptual Measurement by Melfilterbank/Barkfilterbank
- Logarithmic Spectrum Computation
- Feature Generation by Discrete Cosine Transform (DCT)

Other features are obtained by referring to different kinds of given features and scales. We start with the simple approach of the LPC features.

21.3.2.2 LPC Features and Linear Prediction in Speech

Linear Prediction (LP) been studied in Chap. 9 of Part II. Now this study is extended to speech recognition where it plays a major role (Fig. 21.7).

The basic idea of this technique is to obtain the auditory spectrum and then to obtain the features. In the liner prediction coefficients (LPC) feature extraction the signal is segmented and windowed by using Eq. 21.5. The linear prediction (LP) analysis is used to obtain LPC coefficients. It is ruled by the Eq. 21.5 where $s[n]$ denotes the speech signal.

$$s[n] = \sum_{i=1}^p a_i s[n-i] + g_s u_s[n] \quad (21.5)$$

In this equation the signal is obtained as a linear combination of the past p values. These coefficients a_i are unknown. The other parameters of the Eq. 21.5 denote; u_s —a source of the sound (e.g., periodic pulses for vowels, noise for fricative, pulse for plosives) powered by a constant gain g_s . Using this equation gives the LPC coefficients that describe the speech features. The LPC feature extraction technique is shown in Fig. 21.8.

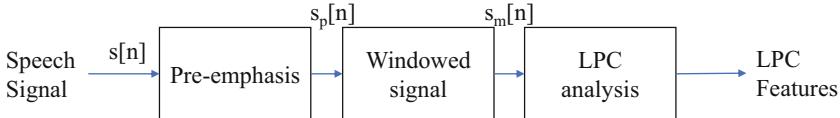


Fig. 21.8 Block diagram of LPC feature extraction

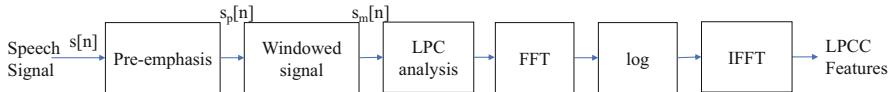


Fig. 21.9 Block diagram of LPCC feature extraction

21.3.2.3 LPCC Features

The extended version of linear prediction is called LPCC which stands for Linear Predictive Cepstral Coefficients, and the Bark scale. The LPCC feature extraction technique is combination of the LPC analysis and the cepstrum analysis that was discussed in Sect. 21.3.2.2. The LPC methods is also know by the names: all-pole model or the auto-regressive (AR) model. It has a good and intuitive interpretation both in time domain and in frequency domain. The basic idea of this technique is the same as in the LPC analysis. In this technique, the windowed signal is used for computing the auto-correlation and the power spectrum. The inverse DFT of the logarithm of the power spectrum is computed for the LPCC features. A block diagram of the LPCC feature extraction technique is shown in Fig. 21.9.

21.3.2.4 PLPCC Features

The Perceptual Linear Prediction Cepstral Coefficients (PLPCC) feature can be used for phonetic segmentation task. First we provide a basic description of PLPCC's. The Fourier transform is applied to compute the short-term power spectrum, and the perceptual properties are applied while the signal is represented in a filter bank. The spectrum is transformed to a Bark scale, and this spectrum is pre-emphasized by a function which approximates the sensitivity of human hearing at different frequencies. The output is compressed to approximate the nonlinear relationship between the intensity of a sound and its perceived loudness. The all-pole model of LP is then used to give a smooth, compact approximation to the simulated auditory spectrum, and finally the LP parameters are transformed using recursive relations to cepstral coefficients for use as segmentation features. The Fourier transform is applied to compute the short-term power spectrum, and the perceptual properties are applied while the signal is represented in filter bank. The spectrum is transformed to a Bark scale, and this spectrum is pre-emphasized by a function which approximates the sensitivity of human hearing at different frequencies. The output is compressed to approximate the nonlinear relationship between the intensity of a sound and its

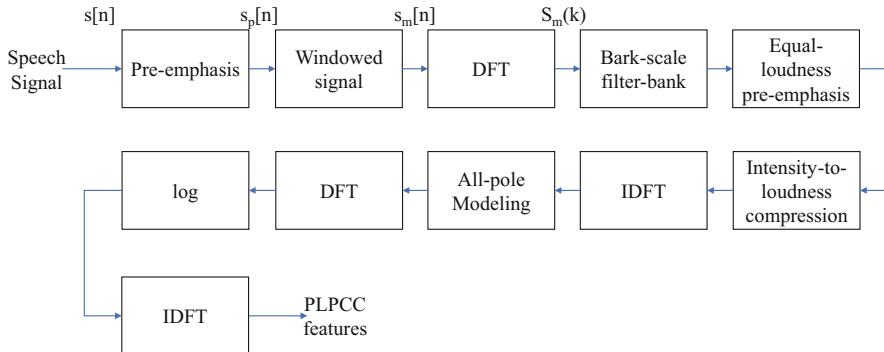


Fig. 21.10 Block diagram of PLPCC feature extraction

perceived loudness. The all-pole model of LP is then used for an approximation to the auditory spectrum. The LP parameters are transformed to cepstral coefficients for use as segmentation features. In a block diagram of this process is shown in Fig. 21.10.

21.3.2.5 SILTT Features

The local trigonometric transformation (LTT) is the basis to extract shift invariant local trigonometric transform (SILTT) features. The idea of this technique is to build a library of functions of an orthonormal basis that compared to the given signal or collection of signals that has the lowest information cost. The SILTT is an adaptive feature extraction technique. This contains multiple representations of the signal divided successively into smaller blocks and expanded into local bases in each block. The state of the art is also comparable to the classic MFCC features. The MFCC consists of windowing the signal applying a window function and a folding operation, which results in disjoint intervals, applying the discrete cosine transform IV (DCT-IV) for spectral analysis, and extracting features based on the cost function which is the spectral entropy of the spectral analysis. Then the inverse discrete cosine transform which is DCT-IV is computed following an unfolding operation. We have divided the section into spectral shaping and spectral analysis. In SILTT, the perceptual feature transformation is not used. A basis is the shift invariant local trigonometric transform (SILTT) features. We discuss computation of these features first. The idea of this technique is to build a library of functions of an orthonormal basis that compared to the given signal or collection of signals has the lowest information cost. They characterize the behavior of the random variables and are quite often useful to estimate the probabilities of events to find the hypothesis of the smallest error. One first selects the window width. Then the procedure is repeated to construct the best basis from all possible local cosine and sine bases using the cost function. For the search for the local cosine and sine basis the best

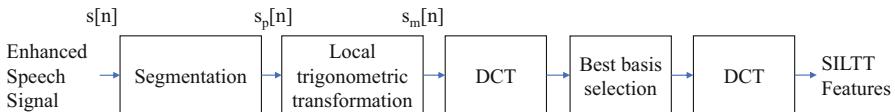


Fig. 21.11 Block diagram of SILTT feature extraction

matching to the signal in terms of the cost function as defined based the entropy of the decomposed signal is selected. The entropy then gives the SILTT features. There are several possible ways to compute the cost function. Examples are Shannon entropy, Neumann entropy, wavelet entropy etc. The Shannon spectral entropy is a common such cost function. In Fig. 21.11 provides a view of the complex SILTT feature extraction.

The SILTT has been used for speech processing and recognition (clean and noisy), the wheezing lung sounds analysis, seismic data processing, among others.

21.4 Background Information

General

A general and recommended source is Young et al. (2005). Basics on the sequence comparison are provided in David and Kruskal (1999). Feature is an expression that in everyday language is used for a property of an object. In signal processing this term is used in more restricted sense. Signal processes have many properties that are not handled easily because of complexity reasons. A feature vector is a short version of such properties. Such feature extraction are elaborated in this book. More detailed information about the feature computation using DCTs and the related transformation can be found in Yip (2000). The comparison between the features LPCC and MFCC is shown in Bhattacharjee (2013).

Past and Suggested

Backpropagation neural nets have been used for learning noise reduction in Tamura and Waibel (1988). It was applied for stationary and for non-stationary distributions. Back propagation is categorized as a supervised learning. Sphinx is a real time large vocabulary, speaker independent speech recognition toolkit, developed at Carnegie Mellon University. Sphinx (2006) uses vector quantization (VQ) based discrete hidden stochastic models. The acoustic models in this toolkit consist of a set of generalized phonetic models designed to be extensible to other applications and for co-articulatory behavior. It is written both in ANSI C and Java. A commercial product is IBM Via Voice (2006) as well as Google speech recognition and tool <https://cloud.google.com/speech-to-text>, Microsoft's. Some recommended readings are Henrique (1999), and Davis and Mermelstein (1980) used in this chapter as references.

<https://www.voicetyper.com>,
Amazon's <https://www.amazon.com/speech-recognition-software>,
Apple's <https://www.macintoshhowto.com/speech-recognition/>,
Nuance <https://www.nuance.com/dragon.html> etc.

21.5 Exercises

Exercise 1 Describe the adaptation steps for SILTT in detail.

Exercise 2 Give an example of an LPC model described in Sect. 21.3.2.2.

Exercise 3 Give a motivation for looking at backpropagation as a form of supervised learning.

References

- [Yip2000] Yip, P.C (2000): The Transformation and Applications Handbook, CRC Press, 2nd Ed, February,
- [David1999] David, S., J. B. Kruskal (1999). Time warps, string edits, and macro molecules: The Theory, a practice of sequence comparison. Center for the Study of Language and Information, Stanford University, 1999.
- [Tamura1988] Tamura S., Waibel A.(1988) Noise reduction using connectionist models. International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1988.
- [Sphinx2006] Sphinx (2006): <http://cmusphinx.sourceforge.net>, October 2006.
- [IBMViaVoice2006] IBM Via Voice (2006). <http://www-4.ibm.com/software/speech/>.
- [Young2005] Young, S., Gunnar, E. and et. al. (2005). The HTK Book, version 3.3. Microsoft Corporation and Cambridge University, UK, 2005.
- [Henrique1999] Henrique S. Malvar. Extended Lapped Transforms: Properties, Applications, and Fast Algorithms, IEEE Transactions in Acoustics, Speech, and Signal Processing, November 1992
- [ch21:DavisandMermelstein1980] Davis, S. and Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech and Signal Processing, 28, 357–366. 1980 <https://doi.org/10.1109/TASSP.1980.1163420>
- [Bhattacharjee2013] Bhattacharjee, R. A Comparative Study Of LPCC And MFCC Features For The Recognition Of Assamese Phonemes, International Journal of Engineering and Technical Research, 2013

Chapter 22

Hidden Stochastic Model for Speech



Overview

The Hidden Markov Model (HMM) has been widely used for pattern recognition since 1970s. One common application of HMM is the acoustic modeling. The HMM has a set of states, a set of transition probabilities from each state to following states and a set of observation probabilities for each state. Given a set of acoustic feature vectors, and word sequence, the most likely word sequence is computed applying Bayes' theorem. The HMM is based on Bayes' principle. For computing the most likely probability, the HMM first computes the likelihood using forward algorithm, then most probable state sequence is computed using Viterbi algorithm, finally the estimation of parameters are obtained by Expectation-Maximization (EM) algorithm. These are discussed in this chapter.

22.1 Generals

We introduced Hidden Stochastic Model earlier. Here we discuss the speech as a special stochastic process. For speech the Hidden Stochastic Model is quite central and it is frequently used. The main two types of variations in speech stochastic processes are the variation in the spectral composition and the variation in the time-scale or the articulation rate. Hidden Stochastic models (HSM) express these variations in the model by the state observation and the state transition probabilities. The speech features are the input to HSM and the output is speech. The basic approach to the recognition problem is to find the most likely the state sequence for a given observation of speech. Each state of an HSM is realized as a model of a segment. The state transition probabilities provide a mechanism for the connection of various states, and for modelling the variations in the duration and time-scales of the signal in each state. For example, the short or slow articulation is accommodated by self-loop transitions in the states of the model where the fast

speaking or articulations are accommodated by skipping next state connection. The state observation probabilities or the probability density functions model the space of the probability distributions of the spectral composition of the signal segments associated with a state. In the presented speech recognition approach, the speech is, as mentioned, spoken command which is a word or set of short words. Some assumptions and concepts listed below are considered here. These assumptions are applicable in solving our recognition problem.

We are taking the view that speech introduces “noise” which makes it hard to recognize the “true” string of words. Our goal is then to build a model of the channel, that introduces that noise, so that we can figure out how it modified this “true” sentence and hence recover it. “Noise Channel” view absorbs all variability’s of the speech mentioned earlier including true noise. Having insight of the noisy channel model means that we know how the channel distorts the source, we could find the correct source sentence for a waveform by taking every possible sentence in the language, running each sentence through our noisy channel model, and seeing if it matches the output. We then select the best matching source sentence as our desired source sentence. Note that making hypothesis about the “what” and “that” hypothesis defines the difficulty of the overall solution.

Implementing the noisy-channel model as we have expressed it in previously requires solutions to two problems.

- (i) First, in order to pick the sentence that best matches the noisy input we will need a complete metric for a “best match”.

Because speech is so variable, an acoustic input sentence will never exactly match any model we have for this sentence. As we have suggested in previous chapters, we will use probability as our metric. This makes the speech recognition problem a special case of Bayesian inference, a method known since the work of Bayes (1763).

- Bayesian inference or Bayesian classification was applied successfully by the 1950s to language problems like optical character recognition (Bledsoe and Browning 1959) and to authorship attribution tasks like the seminal work of Mosteller and Wallace (1964) on determining the authorship of the Federalist papers.
 - Our goal will be to combine various probabilistic models to get a complete estimate for the probability of a noisy acoustic observation-sequence given a candidate source sentence. We can then search through the space of all sentences, and choose the source sentence with the highest probability.
- (ii) Solving the decoding (or search problem) requires efficient search. Since the set of all English sentences is huge, we need an efficient algorithm that will not search through all possible sentences, but only ones that have a good chance of matching the input. Since the search space is so large in speech recognition, efficient search is an important part of the task.

22.2 Hidden Stochastic Model

In HSM modeling it uses information theoretic approach to Automatic Speech Recognition (ASR). Information theoretic approach relies on probabilities. Hence, the goal of the probabilistic noisy channel architecture for speech recognition can be stated as follows: What is the most likely sentence out of all sentences in the language \mathcal{L} given some acoustic input \mathbf{O} ? We can treat the acoustic input \mathbf{O} as a sequence of individual “symbols” or “observations”: For example by slicing up the input every 10 ms, and representing each slice by floating-point values of the energy or frequencies of that slice. Each index then represents some time interval, and successive o_i indicate temporally consecutive slices of the input (note that capital letters will stand for sequences of symbols and lower-case letters for individual symbols): If $P(\mathbf{W}|\mathbf{O})$ denotes the probability that the words \mathbf{W} were spoken, given that the evidence \mathbf{O} was observed, then the recognizer should decide in favor of a word string \mathbf{W} satisfying:

$$\hat{\mathbf{W}} = \underbrace{\operatorname{argmax}}_{\mathbf{W} \in \mathcal{L}} P(\mathbf{W}|\mathbf{O})$$

The recognizer will pick the most likely word string given the observed acoustic evidence. Recall that the function $\operatorname{argmax}_x f(x)$ means “the x such that $f(x)$ is largest”. Equation is guaranteed to give us the optimal sentence \mathbf{W} ; We now need to make the equation operational. That is, for a given sentence \mathbf{W} and acoustic sequence \mathbf{O} we need to compute $P(\mathbf{W}|\mathbf{O})$. From the well known Bayes’ rule of probability theory:

$$P(\mathbf{W}|\mathbf{O})P(\mathbf{O}) = P(\mathbf{O}|\mathbf{W})P(\mathbf{W})$$

$$P(\mathbf{W}|\mathbf{O}) = \frac{P(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{O})}$$

where

- $P(\mathbf{W})$ —Probability that the word string $P(\mathbf{W})$ is uttered.
- $P(\mathbf{O}|\mathbf{W})$ —Probability that when \mathbf{W} was uttered the acoustic evidence \mathbf{O} will be observed.
- $P(\mathbf{O})$ —is the average probability that \mathbf{O} is observed:

$$P(\mathbf{O}) = \sum_{\mathbf{W}} P(\mathbf{O}|\mathbf{W})P(\mathbf{W})$$

Since Maximization in:

$$\hat{\mathbf{W}} = \underbrace{\operatorname{argmax}}_{\mathbf{W} \in \mathcal{L}} \frac{P(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{O})}$$

is carried out with the variable \mathbf{O} fixed (e.g., there is no other acoustic data save the one we are given), it follows from Baye's rule that the recognizer's aim is to find the word string $\hat{\mathbf{W}}$ that maximizes the product $P(\mathbf{O}|\mathbf{W})P(\mathbf{W})$, that is

$$\hat{\mathbf{W}} = \underbrace{\operatorname{argmax}_{\mathbf{W} \in \mathcal{L}}}_{\mathbf{W} \in \mathcal{L}} \frac{P(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{O})}$$

The probabilities on the right-hand side of the last equation presented in previous slide are for the most part easier to compute than $P(\mathbf{W}|\mathbf{O})$. For example, $\mathbf{P}(\mathbf{W})$, the prior probability of the word string itself is exactly what is estimated by the N-gram language models. We will see next that $P(\mathbf{O}|\mathbf{W})$ turns out to be easy to estimate as well. But $P(\mathbf{O})$, the probability of the acoustic observation sequence, turns out to be harder to estimate. Luckily, we can ignore $P(\mathbf{O})$. Why? Since we are maximizing over all possible sentences, we will be computing $P(\mathbf{O}|\mathbf{W})P(\mathbf{W})/P(\mathbf{O})$ for each sentence in the language. The $\mathbf{P}(\mathbf{O})$ does not change for each sentence! For each potential sentence we are still examining the same observations \mathbf{O} , which must have the same probability $P(\mathbf{O})$. Thus:

$$\hat{\mathbf{W}} = \underbrace{\operatorname{argmax}_{\mathbf{W} \in \mathcal{L}}}_{\mathbf{W} \in \mathcal{L}} \overbrace{P(\mathbf{O}|\mathbf{W})}^{\text{likelihood}} \overbrace{P(\mathbf{W})}^{\text{prior}}$$

The language model (LM) prior $P(\mathbf{W})$ expresses how likely a given string of words is to be a source sentence of English. This parameter $P(\mathbf{W})$, is computed using N-gram grammars. However, we will not be discussing language modeling in this chapter.

This chapter will show how the HSM can be used to build an Acoustic Model (AM) which computes the likelihood $P(\mathbf{O}|\mathbf{W})$. Given the AM and LM probabilities, the probabilistic model can be operationalized in a search algorithm so as to compute the maximum probability word string for a given acoustic waveform.

The next descriptions deal with a number of technical details. HSM is comprised of several states. Several states may represent a word. The states for the word may correspond to an HSM model. For example a spoken command "Oeffne" may have 5 states. Further details of the operationalization HSM speech recognizer as it processes a single utterance. The recognition process can be divided in three stages.

- Feature Extraction
- Acoustic Modeling
- Language Modeling

In the feature extraction or signal processing stage, the acoustic waveform is sampled into frames. Each frame is usually of 5, 10, 15, 20, 25 or 30 ms duration.

Thus a frame is represented by a number of speech features in a feature vector. Each time window is thus represented by a vector of around 39 features: $13 + 13 + 13 \Rightarrow$ static + first + second derivatives representing spectral information as well as information about energy, and spectral change (dynamic features). The recognition may be preceded following the frames and can be set up to represent a state. The transition matrix indicates the transition probability of the transmissions from one state to another state. If the HSM model is a 5 state model then the transition matrix will be a 5×5 dimensional matrix. The initial state indicates where to start. The sequential series of states, which may be self-looped, is left-right transitioned. Generally a state might be seen as an HSM model and a connection of the states links the HSM to build a word or words or sentences. Each frame has a set of features in a vector space. A number of frames can be realized as states in the feature space. These features may be realized as Gaussian mixtures models for the HSM in order to consider continuous HSM models. Tracing through the HSM determines the likelihood of spoken utterances and state sequences.

In order to better understand the formations presented in the next section the following introduces the notations and notational conventions.

- **N**: number of states in the model.
 - Set of states $\mathbf{Q} = \{ \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N \}$
 - State at time t , $\mathbf{q} \in \mathbf{Q}$
- **M**: number of observations defined with observations \mathbf{O} , drawn from a vocabulary \mathbf{V} .
 - Observations set $\mathbf{O} = \{ \mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_m \}$
 - Vocabulary $\mathbf{V} = \{ \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m \}$
 - Observation at time t , denoted by $\mathbf{o}_t \in \mathbf{V}$
- $\mathbf{A} = \{a_{ij}\}$: state transition probability distribution matrix.
 - $a_{ij} = P(\mathbf{Q}_{t+1} = \mathbf{q}_j | \mathbf{Q}_t = \mathbf{q}_i) \quad 1 \leq i, j \leq N$
- $\mathbf{B} = \{b_j(\mathbf{o}_t)\}$: a set of observation symbol probability distribution also called emission probabilities.
 - $b_j(\mathbf{o}_t) = P(\mathbf{o}_t | \mathbf{q}_j) \quad 1 \leq j \leq N$
- $\pi = \{\pi_i\}$: Initial state distribution.
 - $\pi_i = P(\mathbf{q}_i) \quad 1 \leq i \leq N$
- A special **start** and **end** state which are not associated with observation:
 - \mathbf{q}_0 and \mathbf{q}_e

The HMM model is typically written as: $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$. This notation also defines/includes the probability measure for \mathbf{O} , i.e., $P(\mathbf{O}|\lambda)$ (Fig. 22.1).

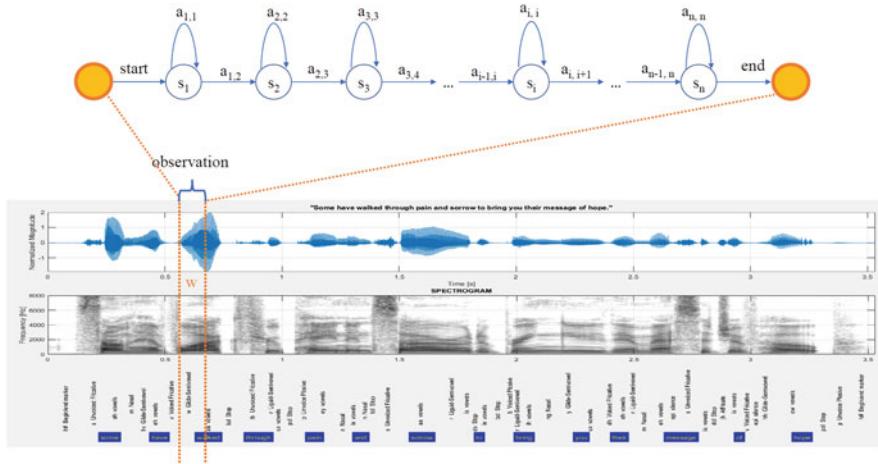


Fig. 22.1 Signal modeling

Example of HSM for Speech Recognition

A three state HMM is presented below: Typical question arises: what are states used for and what do they model? For speech the (hidden) states can be:

- Sub-phones,
- Phones,
- Parts of Speech, or
- Words.

Observation is information about the spectrum and energy of waveform at a point in time. Decoding process maps this sequence of acoustic information to phones and words. The observation sequence for speech recognition is a sequence of acoustic feature vectors. Each acoustic feature vector represents information such as the amount of energy in different frequency bands at a particular point in time. For now we'll simply note that each observation consists of a vector of 39 real-valued features (see previous section) indicating spectral information. Observations are generally drawn every 10 ms, so 1 s of speech requires 100 spectral feature vectors, each vector being of length 39 (static feature 13 + first order derivative features 13 + second order derivative features 13). The hidden states of HSM can be used to model speech in a number of different ways. For small tasks, like **digit recognition**, (the recognition of the 10 digit words: zero through nine), or for **yes-no** recognition (recognition of the two words yes and no), we could build an HMM whose states correspond to entire words. For most larger tasks (e.g. longer words), however, the hidden states of the HSM correspond to phone-like units, and words are sequences of these phone-like units (Fig. 22.2).

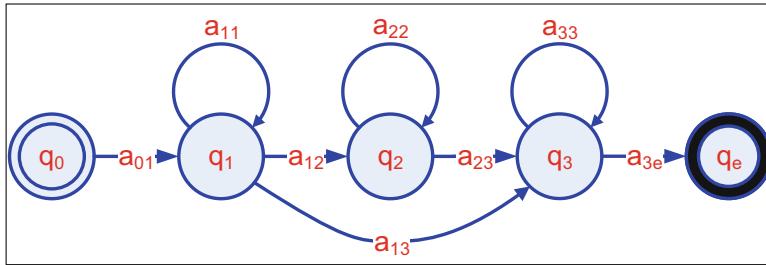


Fig. 22.2 An hypothetical example of a 3 state HMM

22.3 Forward and Backward Predictions

Now we come to the computation of the predictions. This depends of course on what we are predicting. We continue the discussion started in Chap. 9. For completeness we repeat several formulas. Suppose we have a stochastic process $s[n]$. If we look at any time instance t we have two time segments:

- $n, n < t$ The time points before t
- $n, n > t$ The time points after t

If we take some p we can estimate $s[n]$ in two ways:

- Using the p time points before n : **Forward Prediction (FLP)**.
- Using the p time points after n : **Backward Prediction (BLP)**.

For both estimates, forward and backward, we can take any method, for instance the linear prediction.

22.3.1 Forward Algorithm

Forward search is done by using the forward algorithm. In the evaluation, given the model the probability of the set of observations in a specific state sequence is estimated. Given the model, the probability of the state \mathbf{o}_t being at state i at time t is estimated by the forward probability denoted by α . The forward probability can in principle be computed by the recursion by using equation presented below.

$$\alpha_t(i) = p(\mathbf{o}, \mathbf{q} | \lambda) = p(\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t, \mathbf{q}_t = \mathbf{s}_i | \lambda) \quad (22.1)$$

The forward algorithm estimates the likelihood $p(\mathbf{o} | \lambda)$ (which means for the given model λ the probability of the observation \mathbf{o} at certain time at certain state) by the following three steps:

Initialization

The initial parameters to start the evaluation uses the formula presented in the Eq. 22.2:

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1) \quad \text{for } 1 \leq i \leq N \quad (22.2)$$

where we interpret it as initially at time $t = 1$ we are in state q_1 with probability π_{q1} , and generate the symbol/vector \mathbf{o}_1 (in this state) with probability $b_{q1}(\mathbf{o}_1)$. Assuming that we start from only one initial state, and that state is the first state then $i = 1$, the following equation applies:

$$\alpha_1(i = 1) = 1 \quad (22.3)$$

Induction

The probabilities of the observations from the past to the present state are computed using the previous probabilities, transition probabilities and observation probabilities in order to estimate the likelihood of the observations. The computation is shown in Eq. 22.4 for $t = 1, 2, 3, \dots, T$ and $j = 1, 2, 3, \dots, N$:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(\mathbf{o}_{t+1}) \quad 1 \leq j \leq N \quad \& \quad 1 \leq t \leq T \quad (22.4)$$

Termination

The forward probability provides the estimate of the observation for a state given the model, where T is the length of each feature observed sequence. The forward probability α is terminated using Eq. 22.5:

$$\alpha_T(N) = p(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (22.5)$$

where $\mathbf{O} = \mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t, \dots, \mathbf{o}_T$.

22.3.2 Backward Algorithm

Similarly to forward algorithm, let us define the backward variable, $\beta_t(i)$. This parameter is defined as the probability of the partial observation sequence from time $t + 1$ to the end, given state s_i at time t and the model, i.e.

$$\beta_t(i) = P(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \dots, \mathbf{o}_T, q_t = s_i | \lambda) \quad (22.6)$$

The backward probability β denotes the probability of the observations \mathbf{o}_T through \mathbf{o}_{t+1} being in state i at time t given a HMM model.

Here the probability of the future sequence conditioned on the present state i at time t is computed.

Initialization

The β is initialized using Eq. 22.7. Probability of the initial state i at time T is 1.

$$\beta_T(i) = 1 \quad \text{for } 1 \leq i \leq N \quad (22.7)$$

Induction

The likelihood of the observation given the model is shown in Eq. 22.8

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j) \quad T-1 \geq t \geq 1 \quad \& \quad 1 \leq i \leq N \quad (22.8)$$

Termination

The transition of the observations to the end is finished at time $t = 1$. Probability is thus obtained:

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^N \pi_i b_i(\mathbf{o}_1) \beta_1(i) \quad (22.9)$$

22.4 Forward-Backward Prediction

In this approach, one combines the predictions that we just introduced. This prediction can be done in several steps as shown in Fig. 22.3

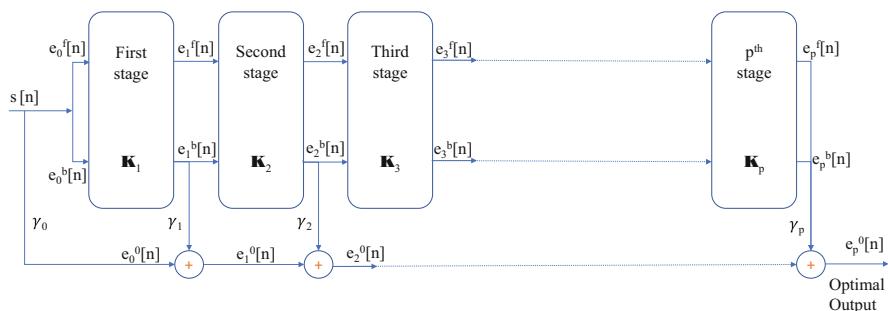


Fig. 22.3 Forward-backward prediction

22.5 Burg Approach

The Burg approach is an order recursive least-squares linear predictor. There order recursive means that if the model is of p th order, we can compute the model parameters of the model order $p + 1$. The auto-correlation and co-variance approaches are fixed order algorithms meaning that they are not order recursive. This says that if we change the order, we need to repeat the whole computation. The Burg approach uses both the forward and backward error minimization approach. We have introduced the forward prediction and its error, now we will introduce the backward prediction (BP) and its error. The order recursive algorithm interconnects the optimum filtering and the FLP and the BLP problems. The optimum filtering refers to the system which response is closest to the desired response. The Burg approach needs to consider the time instance n and the order p such that $i = 1, 2, \dots, p$. Results of the Burg approach are shown in Fig. 22.4. Some terminologies of the Burg approach as the excitation, the speech input and the forward and backward prediction errors are shown in Fig. 22.4. In Fig. 22.5 we observe how the forward and backward predicted values are estimated from the same observation using the same number of samples. We name now the forward prediction error $e[n]$ as $e_f[n]$ for an easier manipulation and it is written in Eq. 22.10 where f denotes forward prediction.

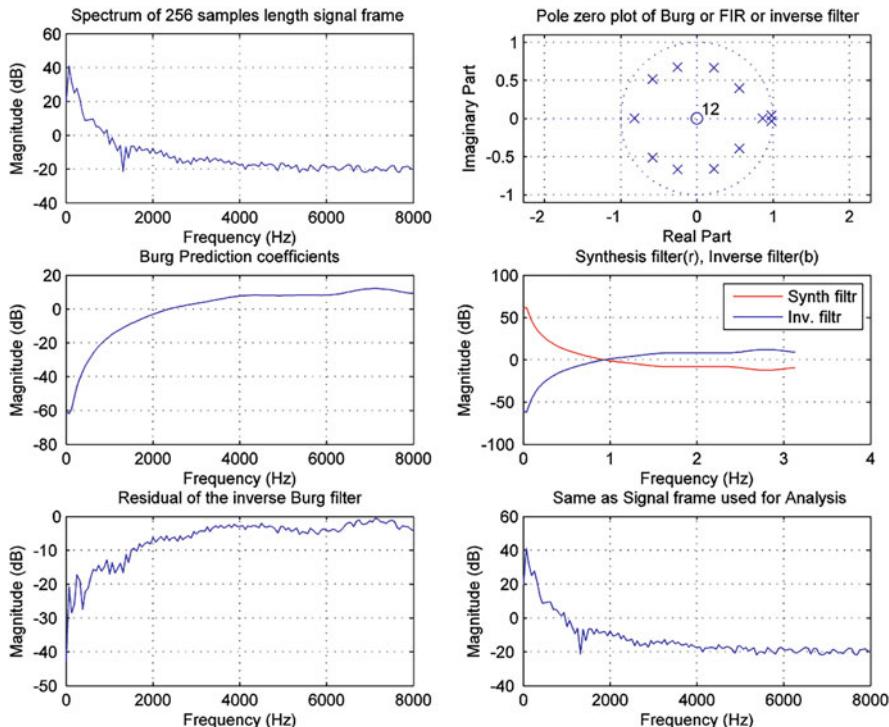


Fig. 22.4 The Burg approach

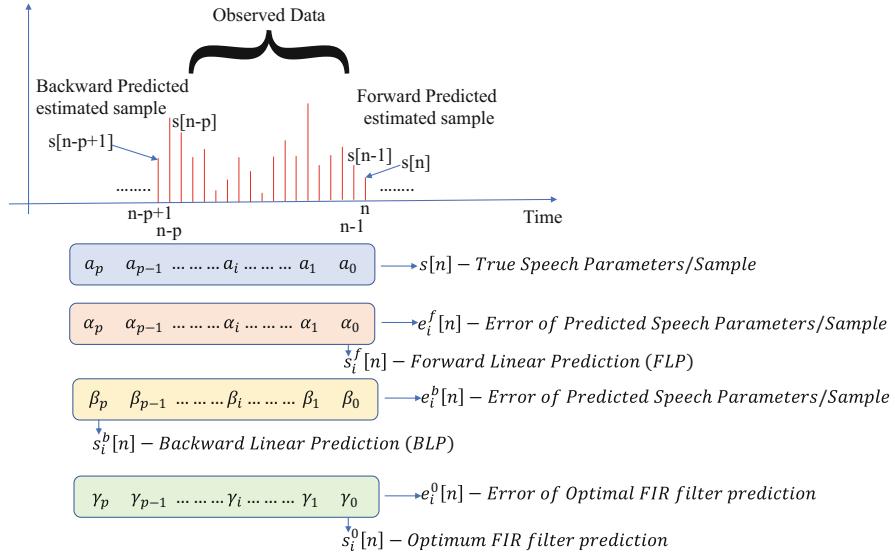


Fig. 22.5 Visualization of forward and backward prediction

$$e^f[n] = s[n] - \hat{s}[n] = s[n] - \sum_{i=0}^p \alpha_i s[n-i] \quad (22.10)$$

For backward prediction the following applies:

$$e^b[n] = s[n] - \hat{s}[n] = s[n] - \sum_{i=0}^p \beta_i s[n-i] \quad (22.11)$$

The consequence is that the spoken words are corrupted dynamically. That means (among others) that the observed classes can overlap to some degree. The spoken words are represented in the 2-dimensional plane. Notation in and the process is depicted in Fig. 22.5:

- (i) Forward prediction: $s[n]$ is to be predicted.
- (ii) Backward prediction: $s[n-p]$ is to be predicted.
- (iii) a_i denote the parameters to be estimated.

Now we refer to Fig. 22.6. The reference words to which we want to map the (corrupted) words are w_1, \dots, w_8 . The undotted lines show the true classes of the (corrupted) spoken words. The dotted lines show the classification obtained by a similarity measure. Because this measure classifies incorrectly the dotted classes

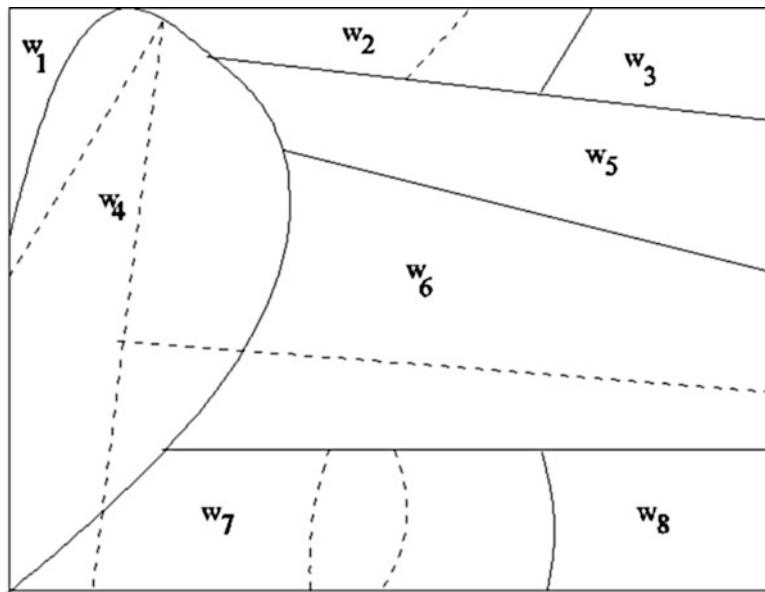


Fig. 22.6 Visualization of hypothetical incorrect classification

are not the same as the true classes. We also see that the dotted classes can overlap as in the cases of w_7 and w_8 . This leads finally to a misclassification.

22.6 Graph Search

Search was introduced in Part II. Now we look at special applications. The first one is concerned with graphs. Problem solving can often be described with a directed labelled graph G which gives an instance of the general search model:

- The nodes of G are labelled with sub problems (or indexes or codes of them).
- The edges are labelled with operations which can be used for problem solving. We assume here that the graph has no cycles.

Graph search in general means to find a node or a path in the graph having a certain property. Solving a problem then reduces to search a solution that constitutes the specific set of nodes in the graph. This can also be done if there are solutions with different quality: One searches for an optimal solution. This solution can, at least in principle, be obtained by searching. However, one has to overcome the problem of the complexity of the search for any methods. Intelligent search uses knowledge. Intelligence means in principle to cleverly find a path in the graph that shortens the search.

22.6.1 Recognition Model with Search

Now we look at search in recognition. There are many different recognition models. For the recognition in the HSM we take here a form defined in the “Dynamic Automatic Noisy Speech Recognition System” DANSR. Figure 22.7 visualizes this. The solution finding is represented as a graph search. The model combines different elements. For the acoustic model a forward-backward algorithm is used. The language model uses a constrained Viterbi algorithm. This also applied to find the most likely word in the vocabulary. Next we give an overview over a complex system of speech recognition in presence of noise. In Figs. 22.8 and 22.9, we show the illustration of the DANSR system. This is our example system and was developed by one of the authors. It contains many of the methods described already. It is a very complex system and it also handle variety of noise types.

A more detailed description is provided next. The diagram itself is complex because it describes a complex situation due to the possible presence of different kinds of noise that, however, do not always occur at the same time. The first vertical line describes the language generation in the human body. However, we will not discuss it here. The other big triangle contains the system itself. One sees the different methods and for which tasks they are used. The rectangle contains the elements of the system.

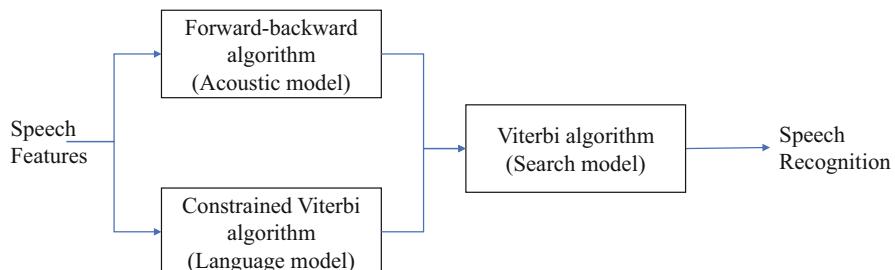


Fig. 22.7 HSM using hidden Markov modeling (HMM) for acoustic and language search in recognition

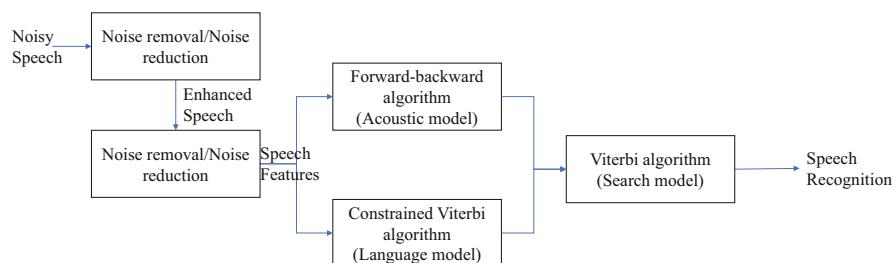


Fig. 22.8 HSM in DANSR model definition

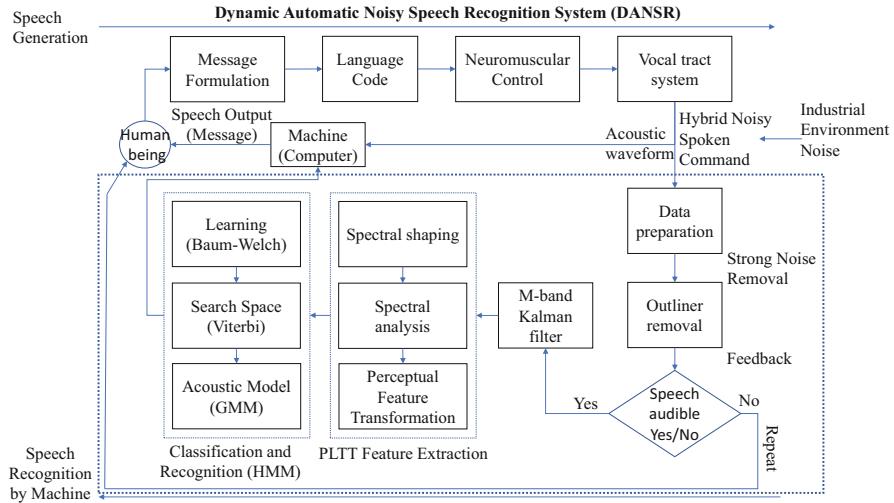


Fig. 22.9 The DANSR system

Application: Speaking in a Car

Typically, a manual control input enables the speech recognition system and this is transferred to the driver by a prompt. Following the audio prompt, the system has a “listening window” during which it may accept a speech input for recognition. Simple voice commands may be used to initiate phone calls, select radio stations or play music from a compatible smartphone, MP3 player or music-loaded flash drive. Voice recognition capabilities vary between car make and model. Some of the recent car models offer natural-language speech recognition in place of a fixed set of commands allowing the driver to use full sentences and common phrases. With such systems there is, therefore, no need for the user to memorize a set of fixed command words.

22.7 Semantic Issues and Industrial Applications

So far, we were dealing with syntactic aspects of a language only. That means we did not assign a meaning to the digital information. However, this is the most important issue in speech understanding. Now we discuss this from a simple principal (not linguistic) point of view. When a machine receives a message it is usually expected that it performs a certain action. We call this the semantics or the meaning of the message. Such an action can be the formulation of another message or a handling that manipulates the environment. For this reason we call the first message a command. In order to perform the intended action another machine called execution

machine is needed. We assume that if this machine obtains the command it executes the intended action. Therefore the semantics is twofold:

- (i) Recognition of the spoken command
- (ii) Execution of the command.

For the recognition of the spoken command it is sufficient to have a list of words in the vocabulary. If a word from this list is spoken that will not be presented in a clear form then it cannot have a meaning. There are different possibilities:

- (i) There is the speaker as an external participant. The speaker utters the words.
- (ii) The system has the following components:

The

- (a) microphone,
- (b) signal analyzer, and
- (c) deioniser,

the latter two are not shown in the figure.

- (iii) The recognition component accepts spoken language that it has to understand.
- (iv) The analysis component obtains the spoken phrase as input e.g. in form of an abstract command. It analyzes the commands with respect to correctness, errors, plausibility and related aspects and thus generating an understanding of the spoken phrase as far as necessary. It can give a spoken feedback to the speaker if this is needed for the understanding of the spoken words.
- (v) The execution component performs the action intended by the human. It has a (formal) representation of the actions and processes.

Figure 22.10 shows the overall system were noise is absent. A simple approach uses the fact that one can completely describe the scenario, i.e., we have a closed world scenario. This allows us to discuss all problems internally. There is, however, one part of the relevant world that is not represented in the scenario; this comprises the speaker and the internal states of the speaker. In order to get access to this, we need a communication with the speaker in spoken language. The words in the commands are elements of the vocabulary of the system. A problem is that one cannot utter the commands in exactly the intended formal way. There are two ways to cope with this problem:

- (i) **Top-down approach:** Recognize the sentence with a recognizer that has linguistic capabilities and reconstruct the command with some method that one applies it in a second step. This is the standard in existing systems.
- (ii) **Bottom-up approach:** Recognize only key words from the commands and synthesize the command. Key words are e.g. “open” and “window”. From these key words the command is synthesized.

Both ways assume that the receiver has the ability to perform the command. This requires some technical capacities that needs an additional machinery and expert knowledge.

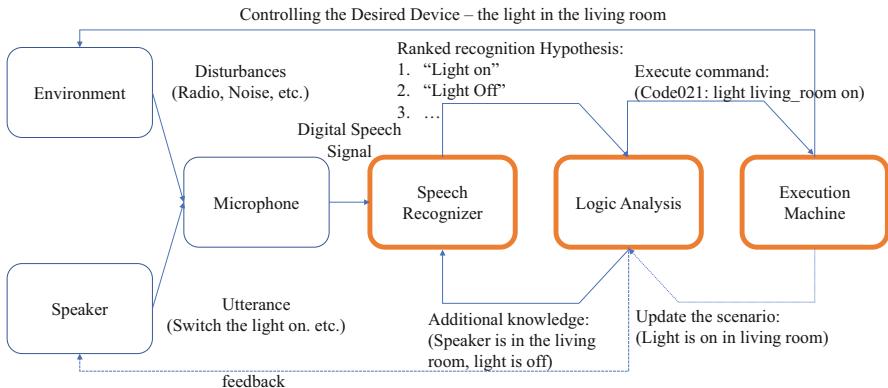


Fig. 22.10 Overall view of ambient assistive living room for a nursing home

22.8 Problems with Noise

An additional but serious problem is the presence of different kinds of noise. Noise disturbs the original signal process. For this we have provided several studies and methods in this book. We provide various methods to deal with all kinds of noises. When semantics is involved as in speech the noise can change the meaning and speech can be interpreted with different meaning than intended. The same happens to bio-medical applications. This is different in music, see below. Here we rely on the methods handling noise that we previously introduced.

22.9 Aspects of Music

The progress in the recent years has increased the understanding how one produces and receives music. This has benefited from the progress in understanding speech. Our view is that music provides sounds and sounds consist of signal sequences. Many aspects in music are of the kind of psychoacoustic phenomena discussed above. A major shift has been that many sound synthesis products that have traditionally been hardware electronics have become software. Audio signals are sound waves—longitudinal waves which travel through the air, consisting of compressions and rarefactions. One measures these audio signals in decibels. Overtones are other sinusoidal components present at frequencies above the fundamental. All of the frequency components that make up the wave constitute the total waveform. All of the variations in air pressure against the ear drum and the subsequent physical and neurological processing and interpretation, give rise to the subjective experience called sound accepted by humans. Mostly one assumes that the peripheral parts of hearing can be modelled by a bank of band pass filters, followed by half-

wave rectification and compression of the sub band signals. Besides the physical aspects of music, there is a symbolic way to record music. This is in analogy to record spoken language in the form of a text. The symbolic representation of music we see in Fig. 22.11. Here we remark that one can express in this representation psychoacoustic elements. It does not include all psychoacoustic elements but several. This gives a composer the task to foresee which elements have which impression on the listener. One cannot represent these elements in written language for speech. The loudness and the applied musical instruments are not specified for individual notes but are determined for larger parts. In general, written music is primarily a performance instruction, rather than a representation of music. It describes music in a language that a musician understands and can use to produce musical sound. Each instrument has a characteristic pattern of waves. The fundamental is the frequency at which the entire wave vibrates. Some examples of the applications of music signal processing methods include the following:

- Music coding for efficient storage and transmission of music signals. Examples are MP3, and Sony's adaptive transform acoustic coder.
- Music sound effects as in 3-D (or 5-D etc.) spatial surround music and special effect sounds in cinemas and theatres.
- Music synthesis, pitch modification, audio mixing, instrument morphing (i.e. the creation of hybrids of different instruments that can be constructed digitally but may not be physically possible), audio morphing (i.e. mixing the characteristics of different vocal or musical sounds), music and vocal signal separation, audio editing and computer music composition.
- Music transcription and content classification, music search engines for the Internet.
- Noise reduction and distortion equalization such as Dolby systems, restoration of old audio records degraded with hiss, crackles etc., and signal processing systems that model and compensate for non-ideal characteristics of loudspeakers and music halls.

Next, we explain some basic terms in the content of music. Most of them have specific forms when considering instruments. Music content creation has a different set of objectives concerned with the creative methods of composition of music content. It is driven by the demand for electronic music instruments, computer music software, digital sound editors and mixers and sound effect creation. In this chapter we are mainly concerned with the transformations and modelling of music signals.

A music interval is the relation between the frequencies of two notes, or the pitches of the notes, at the beginning and the end of the interval. This relation can be expressed as the ratio or the distance of the frequencies (or pitch) of the musical notes at the beginning and the end of the intervals. A musical scale is a specific pattern of the pitch ratios of successive notes. The pitch difference between successive notes is denoted as a scale step. Musical scales are usually known by the type of interval and scale step that they contain. In music one is also looking for models. Besides the often-used probabilistic models one efficient way of structuring the transcription problem is through so-called mid-level representations.

Rondo Alla Turca “Turkish March”
 3rd movement from Sonata K.331

W.A Mozart
 Arr. by supervoice

Fig. 22.11 Musical notation in a score

A fundamental mid-level representation in human hearing is the signal in the auditory nerve. Whereas we know rather little about the exact mechanisms of the brain, there is much wider consensus about the mechanisms of the physiological and more peripheral parts of hearing. There one has the following parameters:

- Sampling frequency. This number is how many samples per second were taken when converting the analog signal to a digital one. It is measured in Hz. Sampling frequency is sometimes also written as f_s .
- The number of data points in the input.
- The total time T we're sampling over. This is measured in seconds.
- Frequency resolution. This is defined as $\frac{1}{T}$.

Those are some of the parameters that are used to define the music sounds.

22.10 Music Reception

Musical tones have three identifying characteristics; volume, pitch and timbre. Volume is power, or the amplitude of the corresponding wave, and it is measured in decibels. Frequency is the measure of how “high” or “low” a tone is, which is measured in hertz. Music can be defined as organised sound comprising the following structural elements: pitch, timbre, key, harmony, loudness (or amplitude), rhythm, meter, and tempo. Processing these elements involves almost every region of the brain and nearly every neural subsystem. The process starts with the brain’s primary auditory cortex receiving signals from the inner ear which immediately activates our brain, the cerebellum. We will not try to understand how our brains process music. This needs in particular to study neural codes. However, this is the same problem as for speech. The human auditory system is the most reliable acoustic analysis tool of today. It is therefore reasonable to learn from its structure and function as much as possible. One more approach to structure is to study the conscious transcription process of human musicians and to inquire their transcription strategies. The aim of this is to determine the sequence of actions or processing steps that leads to the transcription result.

22.11 Background Information

General

For speech we took a simple situation where the speech is a command requiring an action. The receiver is often a machine. The computer system has access to an execution machine which it has to transfer the command for execution the action. An example is described in Paul et al. (2014). For general aspects see Lanman (2005) and Ellis (2005).

Past

Search is involved in speech techniques in several ways. This depends on the property of speech. One can find more on the comparison between Viterbi and DTW search and speech aspects of HSM in Hosom (2011). A complete description

DANSR is given in Paul (2014). MFCC is probably the first perceptual speech feature extraction technique for the speech recognition system.

The Mel scale is first used in ASR systems for perceptual speech feature extraction. There is no single mel-scale formula and several formulas have been proposed, see Davis and Mermelstein (1980). Rabiner (1989) provides an excellent, introduction into Hidden Markov Models. In Rabiner and Juang (1993), Rabiner (1989), one finds a description of the Levinson-Durbin approach. The autonomous search process was first suggested by Klapuri (2004) in Kühlthau (1991). Early discussions of formants are given in Benade (1976). It contains also elements of music.

Suggestion

Music, its creation and its reception are quite different from the speech. In speech one does not say how to pronounce a word but in music there are some elements in the notes that say how to play or how to sing. These aspects are related to the fact that one can express several psychoacoustic elements in the notes and such elements play a serious role in music. If one wants to perform a Fourier transform on a sampled signal one has to use the DFT (Discrete Fourier Transform). The FFT arrives at the same result as the DFT but its run time is worse. For music and signals see Klapuri (2004) and Moorer (1977). See also Brandenburg et al. (2002). The DANSR system is contained in the dissertation Paul (2014).

The readers are encouraged to read reference Vaseghi (2008) used in this chapter.

22.12 Exercises

Exercise 1 Select some formula for the mel-scale and generate a function for frequency.

Exercise 2 Compare Mel Frequency and Cepstral Analysis (MFCC) for speech feature extraction.

Exercise 3 Provide an application with more than one hidden processes.

References

[Hosom2011] Hosom, J.-P. (2011): Speech Recognition with Hidden Markov Models. Lectures Winter 2011. Oregon Health & Science University, Center for Spoken Language Understanding

[Vaseghi2008] Vaseghi S.V. (2008). Multimedia Signal Processing: Theory in Speech, Music and Communications. Wiley, USA.

[Davis1980] Davis S.B., Mermelstein P. (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, IEEE ASSP, No. 4, Vol. 28, pp. 357–366.

[Kuhlthau1991] Kühlthau, C. (1991): Inside the Search Process: Information Seeking from the User's Perspective. Journal of the American Society for Information Science.

- [Ellis2005]** Ellis D.P.W. (2005). PLP and Rasta, and MFCC, and inversion in matlab. Wavepage, 2005.
- [Lanman2005]** Lanman D.R. (2005). Design of a sound level meter. Laboratory Report EN 253: Matlab Exercise.
- [Rabiner1993]** Rabiner L. R. and B.-H. Juang. Fundamentals of Speech Recognition. Prentice Hall, New Jersey, USA, 1993.
- [Rabiner1989]** Rabiner, L. (1989). A tutorial on Hidden Stochastic Models and selected applications in speech recognition. Proc of the IEEE, 77(2), pp. 257–286.
- [PaulMMR2014]** Paul, S., Michel V, Richter, M. M. Reaction to Hybrid Noise in Communication. Noise Pollution: Sources, Effects on Workplace Productivity and Health Implications. Invited Book-chapter, Nova publication, 2013–2014, NY, USA.
- [Brandenburg2002]** Brandenburg, Karlheinz et al. (2002) Applications of Digital Signal Processing to Audio and Acoustics. Kluwer Academic Publisher, Boston, USA, 2002.
- [Paul2014]** Paul (2014): DANSR. Dissertation Kaiserslautern 2014.
- [Benade1976]** Benade, A. H. (1976) Fundamentals of musical acoustics, Oxford University Press, London.
- [Moorer1977]** Moorer, J. A. (1977). Signal processing aspects of computer music: A survey. Proceedings of the IEEE.
- [Klapuri2004]** Klapuri, A. (2004), Signal processing methods for the automatic transcription of music.

Chapter 23

Different Speech Applications—Part A



Overview

Now let us see a number of instructive applications. We structure them into groups. In this chapter we discuss the situations of an experimental laboratory and a hospital. Despite the fact that these applications seem very remote from each other, they have something in common. This is twofold:

- (1) There are commands that have to be performed.
- (2) The speech is in a noisy environment, and that is in the center of interest.

Both will be detailed in this chapter. Further scenarios are given in the subsequent sections. In all these applications, we find different kinds of realizing the semantics; they fit, however, in the principles discussed above.

23.1 Generalities

Applications can be discussed on various levels of abstraction. The lowest level uses signals, this is, the level where the computer accepts signals. That is our starting and main point. The signals describe messages to a computer, as explained above. However, the signals are not understood directly. For this one needs to go to higher levels of abstraction. Therefore we provide some illustrative examples. All of them have been implemented and used in a real world environment.

23.2 Example Applications

We describe applications where commands are given from humans to a machine. We assume to have a fixed finite set of commands where the speaker can choose

from. These commands have to be understood by the machine and it has to organize the answer that is usually in form of an action. These examples need no knowledge in order to understand the technical details and the background of the applications. We present three examples of such communications. In each example one or more of the introduced noise types is disturbing the speech. We want that the commands be performed. This needs some organization because additional agents are involved. This requires that

- (a) The meaning of each command is understood.
- (b) There exist a machine, or a person, or an institution that has to understand the command and organize the demand.

The first problem is to establish the semantics of the command. The semantics is here the intended action by the command. The action can be of different kinds. Main types are:

- An answer in the form of speech or text.
- Some action in the environment that has to be performed. Such an action changes the environment.

However, formal semantic and linguistics details are not considered in these analysis. The solution to the problem is to use the list of commands and to represent them. This includes possible deviations from the command list. In our scenarios the noise types occur simultaneously in an unpredictable way. The noise types are the ones that have been discussed in Chap. 19. We will handle them in a way which we call hybrid. The individual methods are those that have been introduced.

In this chapter we will provide how we first asses the noises and characterize the noise extremity. In particular, we characterize the noise by measuring the sound pressure level (SPL) of the sound, compute the frequency information of the sound, the duration of the sound and the environment of the source and generation. Then we control the noise by applying digital and adaptive filtering. For this we have developed a hybrid speech enhancement model. Whenever a command is uttered we have to find this command in our list. The problem is now that the command is corrupted by the noise. For this we use one of the noise removal techniques that we have introduced. Treating the noise enhances the communication in the noisy environment, increases the work place productivity and improves health care because people are not required to expose very often themselves to the noise and a machine can perform the task where the machine is embedded in a speech recognition system. As one can see the approaches in the experiments have much in common. They follow the principles described in the previous chapters. The differences in the approaches are coming from elements described there. Now we start with the example applications.

23.2.1 *Experimental Laboratory*

In a workshop, a lab or an industrial environment the noise is quite common and it has a rich variety. This can be mild, strong, steady and varying. In our example one gives commands in a laboratory to a machine which automatically performs it. This saves to employ a human what may also be dangerous. There are finitely many commands that may, however, additional parameters have. The parameters also occur in the vocabulary. A specific example of noise is if the person speaks dialect. The noise problem one can remove or at least diminish efficiently and economically by using a speech recognition system. Here the speech recognition works as a replacement of the human listener by a machine.

There are here two types of noise. On the one hand, there is the Steady-unsteady time varying noise. In addition, one has the strong noise. It satisfies the conditions of the outliers. Hence the noise can be handled as shown in Chap. 19. The commands between a human being and a computer can be performed remotely in order to operate the machine. Hence, a human being is not required to expose himself to the noisy environment frequently and thus healthy potentials are preserved. At the same time, it is economically of benefit because one replaces the man power by a computer. The semantic issue is done using a finite list of commands. Each spoken words should be one of those. If it is not in the list it should be ignored. This takes care of some filling words like “please”. Therefore the recognition problem reduces to find the corresponding word in the list. There are several possible speaking humans. Often, they do not speak clearly. The solution is by solving the noise problem first and then recognizing the enhanced speech. Subsequently, one embeds the system into the machine in order to function itself as a replacement for a human being.

In this situation, the speaker is a human and the receiver is a machine. The human speaker is not formally defined and is in addition influenced by noise from the environment. The process has mental as well as physical parts that are located at different levels of abstraction. The mind of the human creates the message where also the verbal formulation takes place. It goes then to various parts of the body and ultimately the speech is send as an acoustic waveform. If a human receives the message the physical part is done again automatically by devices in the human ear and brain. This is not the case when the receiver is a machine; it is the purpose of speech recognition to realize the reception.

Receiving the sounds does not include understanding. The main part of receiving speech is identifying the words. The human is able to identify them even in unclear speech. The human has learned the words in question in the past. In a computer one needs:

- (a) A collection of words or phrases that contain all possibilities that are intended to be spoken. This constitutes a case base in the sense of case-based reasoning.
- (b) A way to match an actually uttered phrase with a given phrase. For this a similarity measure is required for performing a nearest neighbor search.

For recognition by a machine, a long training process is necessary. Therefore, a small vocabulary makes the task easier. We describe the whole architecture in order to give an overview over the approach. The architecture is built in order to realize the intended goals. It has four major components that are organized in a modular way so that they can be improved and/or modularized independently. The system has to perform the following tasks:

Task 1:. Understand a spoken demand or command.

Task 2:. Analyse the understood command with respect to possible errors.

Task 3:. Transform the recognized sentence into a formal and an executable form and send it to an execution machine.

Task 4:. Execute the command.

The architecture has independent modules that are connected by interfaces and communicate with each other:

- The speaker utters the words.
- The system has the following components:
 - The microphone, signal analyser and denier.
 - The recognition component accepts spoken language that it has to understand.
 - The analysis component obtains the spoken phrase as input in form of an abstract command. It analyzes the commands with respect to correctness, errors, plausibility and related aspects and thus generating an understanding of the spoken phrase as far as necessary. It can give a spoken feedback to the speaker if this is needed for the understanding of the spoken words.
 - The execution component performs the action intended by the human. It has a (formal) representation of the actions and processes.

In addition, there is the speaker as an external participant. We have the adopted classical approaches for speech recognition; they are:

- Applying adapted filtering to noise when the noise is extreme.
- Speech features are extracted mainly using Shift Invariant Local Trigonometric Transformation.

For comparison, we also used Mel Frequency Cepstral Analysis (MFCC) for speech feature extraction. Finally, the matching is obtained for a recognition result for the Hidden Stochastic Model (HSM) by using Dynamic Time Warping (DTW).

The execution machine has a device that operates an intended action if it gets a certain command. The new element is that there is no human that initiates this action and the human does not need to understand the working of the machine. A major problem is that there are different kinds of noise that occur simultaneously. Therefore the inputs are usually the desired spoken command and undesired different types of signals such as noise or the different types of environmental impacts. Such noise occurs regularly in a lab with machines. Strong noise occurs if suddenly a compact object is falling down to the bottom. This happens for a short time and at irregular times. Hence, the strong noise meets the assumptions put in

the description. The implementation follows the text, in particular for the strong noise. The implemented system is tested in three companies in Germany (Mainz and Kaiserslautern).

23.2.2 ***Health Care Support (Everyday Actions)***

Here automatic speech recognition (ASR) system is used for assisted living. In a nursing home elderly or disabled old person live who is incapable of executing. Persons who are sick and/or unable to move may want to give commands that a machine automatically executes without calling a nurse. This implies some intellectuality but no physical strength from the persons. The environment of the persons consists of a living room in which there are for instance several lights, tables, chairs, a TV-set and a radio. Furthermore, one should control window blinds and doors. The control should be handled by commands provided by the person in the room. The command can, however, be formulated in several different ways. For instance, the command “open the window” can be formulated as: “open the window”, “open the window now”, “open the window right now”, “please open the window”, “open the window, I need fresh air”, etc. There are several problems involved that prevent us from simply using a commercially available speech recognizer. The most important problems are:

- (a) Existence of unpredictable noise.
- (b) Formulation of an erroneous command (e.g. one that is impossible to perform, dangerous or highly implausible).
- (c) Formulation of an incomplete command and, thus, creation of an ambiguity.
- (d) Utterance of words or phrases not in the vocabulary.
- (e) Interference by other speakers (persons, radio, TV).
- (f) Occurrence of noise (e.g. due to open windows), speech by other persons, TV or radio, etc.

The communication is still done here by using a speech recognition system. For this, we apply the methods introduced so far. We collect data from the place, extract features using the MFCC feature extraction technique and recognize the speech using the Dynamic Time Warping (DTW) technique. The ASR system assists the physically or disabled persons to perform certain physical tasks automatically. For this, the persons can give a spoken command like “open the door”. Then there are two machines that have some interplay:

- (a) A first machine that accepts and understands the command and transforms it to a form understandable by the second machine. It also delivers the command to the second machine.
- (b) The second machine accepts this command and is able to execute it. The execution machine is distributed over the whole room. For instance, it has to operate the door and the windows or to switch the lights. Therefore, the

machine has knowledge about all details of the room and its present states of its parameters. It gives possibly even a feedback. The realization needs engineering capabilities.

This is the same as in the last example of a laboratory. In order to understand the command one does the same as in the last project: The possible commands are in a list together with the action to be performed. This means that the machine has to handle certain variations in the formulations of the commands.

The second machine has a complex character and building it needs engineering capabilities. It contains parts that perform the intended actions like opening a window. For this the machine has commands available and calling the commands the actions are performed. For the connection to the speech recognition, the output of the recognition system serves as the input to the execution machine.

In this scenario, the nursing room is embedded with the ASR system. The user or the physically disabled person utters a command in the room which has to be performed. This gives back a certain aspect of independence to the persons. In addition, it supports the personnel of the institution.

The first goal is the modelling of the room because the commands refer to it. The commands demand to modify parts of the room and the machine needs to know it. The analysis refers to the whole room with the necessary details. The representation has to be flexible in the sense that elements of the room can be changed (e.g. a window can be open or closed). For such a modelling, it is recommended to use a tool. An example is Protégé of the Stanford University which was used in this application. It allows sufficiently complex representations.

In this example one approaches the problem by collecting data according to some predefined list. One extracts features by applying the shift invariant local trigonometric transformation (SILTT) and classifies them in order to be recognized. The recognition is done by applying the DTW pattern matching method. The distinction between training and test phase plays already a role in preprocessing. In the training phase, one repeats the preprocessing for all the collected data in the list. The digitized one dimensional speech signal is then processed into multidimensional but lower quantity than the digitized signal. The collected data have to be managed using data structure implementation so that the data can be available and used smoothly for the testing. In the test phase, the unknown speech or word is collected. This is easier to manage from the data structure point of view. Managing the data for testing is more straightforward than the testing phase but the difficulties to manage the data in a real time for speech recognition system. The real time data captured has to be system portable.

For achieving the second goal, one describes the objects in the scenario by an ontology as seen in Fig. 23.1. The objects are not just physical objects. There one sees also the relations and the possible interplay between the objects. The details of the interplay are not shown.

For dealing with the dynamic character of the objects, the attribute states where the values can be changed by the actions used. For instance, a window has the states open, closed. In addition, the objects may have other parameters that are of interest

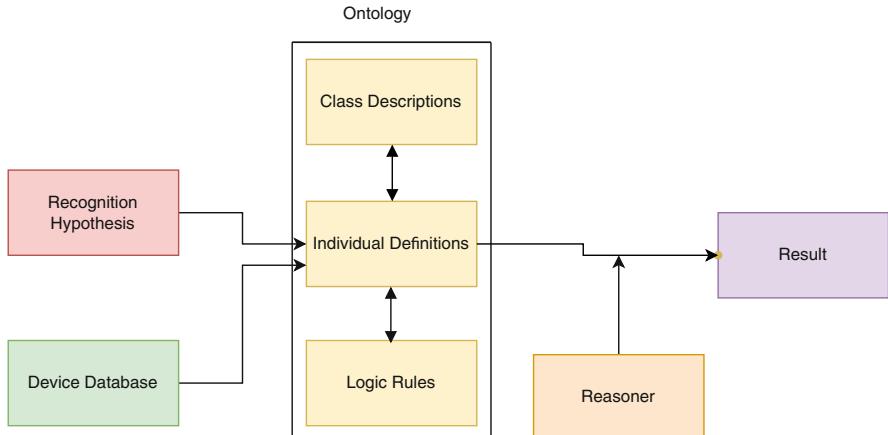


Fig. 23.1 Analysis component of health care support

to the user but cannot be changed. The actions are also indicated in the ontology. One describes them as mappings:

A : states vs states

The actions have the form “action (object)” what says on which object the action is operating, e.g. action (window). For each action we have the following attributes:

- Action operations, e.g. *action (light)_operations* = {from Switch On to Switch Off and vice versa}
 - Action rating = {Possible, Impossible, Critical, Uncritical, Plausible, Implausible}.

To describe this in detail one needs a much more refined ontology. The execution component receives a command from the analysis component. At the same time, it gets a message from the sensor component with additional information. Based on this, the component executes the command in the real world. This may be impossible for some reason that is unknown to the analysis component, e.g. if some part is broken. In each case, the component sends a message to the analyzer, either “Executed” or “Failed” and new states in order to update the dynamic memory.

Let $\psi(i, j)$ be the predecessor grid point (k, l) chosen during the optimization step at grid point (i, j) .

With these definitions, the dynamic programming (DP) algorithm can be written as follows:

Initialization:

- Grid point $(0, 0)$ such that $\psi(0, 0) = (-1, -1)$
 - $\delta_j(0) = d(0, 0)$

Initialize first column for $\{i = 1 \text{ to } T_w - 1\}$

Iteration:

Compute:

For $j = 1$ to $T_x - 1$: swap arrays $\delta_{j-1}(.)$ and $\delta_j(.)$

First point $i = 0$: $\delta_j(0) = d(0, j) + \delta_{j-1}(0)\psi(0, j) = (0, j - 1)$.

Compare column j for $i = 1$ to $T_w - 1$:

Optimization step:

$\delta_{j-1}(0) + d(i, j)\delta_j(i) = \min \delta_{j-1}(i - 1) + 2 \cdot d(i, j)$

Tracking of path decisions

(i) $\varphi(i, j) = \operatorname{argmin}(\delta_{j-1} + d(i, j + 2 \cdot d(i, j))$

(ii) $d(i, j - (-1)) = \delta_j(i - 1) + d(i, j)$

(iii) $\delta_j(k, i) \in \delta(i - 1, j - 1)d(i, j)$

In this application, the system had a certain additional intelligence. It had a complete description of the present room including all actual parameters that are updated after each action. This made it possible to check properties of the execution like the possibility of performing an action. There was a feedback to the user. This enabled the system to ask: “You asked open the door. We have two doors, which one do you mean?” In this application the advantage is the fact that the scenario is completely described, i.e., we have a closed world scenario. This allows us to discuss all problems internally.

The system was developed in a home in Trippstadt, Germany, for old and sick persons and there also tested. The mechanical equipment was performed by the company CIBEK.

23.2.3 Diagnostic Support for Persons with Possible Dementia

Suppose that in a hospital there are many persons with possible dementia. A standard way in medicine to find this out is first to ask them questions like “What is today the day of the week?” There is no specific way to answer and everybody can answer this. Instead one could also ask “What is your first name?” If the answer is wrong then there is a possibility of dementia and a doctor is called for further investigations. If the answer is correct then nothing further will happen. In most cases the answer of the persons will be correct and no medical person has to come. That means a medical person has mostly nothing to do except asking the same question over and over again, i.e. spending unnecessarily much time. The structure of the communications is as follows. First there are four participants:

- The human who ask the questions, using a microphone.
- The second human who receives the question. This person understands the question linguistically and has to answer the question. This answer may not be correct. It sends the answer to the next receiver.
- This receiver is an automaton. It decides about the
- correctness of the answer and tells this to another automaton.

(e) This automaton tells the result of the correctness test to the first human.

There are some difficulties incorporated that we discuss now.

- (1) The second participant is not one person but many that are not under control. They speak differently. It is not possible to perform training with these people and it is not necessary. Instead, we produce the speech with one clearly speaking person. The deviations resulting from speech of other persons are handled as noise. This is possible because there is a very small vocabulary.
- (2) One does only partially know what they speak. One can list some possible answers to the questions. This list is presented to the automaton and learned. Anything outside of this list is considered as a wrong answer.

Now task of the machine is:

- To receive the speech correctly.
- To decide about the correctness of the answer.
- To inform a medical person if the answer is wrong.

This is in the scope discussed so far. Here the term command may be a little misleading. Rather we use the term question. The questions are again in a list together with the correct answers.

In this, project humans and machines cooperate. The machine performs the understanding of the patient and decides about the correctness of the answer and informs the medical expert. If needed this human does the further medical investigations too. The Mel Frequency Cepstral Coding (MFCC) technique was used in the training phase and Dynamic Time Wrapping (DTW) was used in the testing phase. The nearest neighbor selection approach was employed with the DTW technique to find the best match. A comparison is made between four speech recognition approaches: DTW based, hidden stochastic model based, and two commercial recognition software packages, Windows XP, for instance Now and Dragon. A collection of signal processing techniques is used in the front-end. Template matching or stochastic processing are commonly used in the back-end in the development of speech recognition system. Dynamic Time Warping (DTW) is the most popular technique in template matching. Hidden Stochastic Model (HSM) is one of the most commonly used tools in the stochastic processes in solving speech recognition problems. There was no strong noise, it was relatively mild.

The technical steps in detail in the project are as follows

- Capturing speech by a microphone.
- Pre-processing the captured analog speech waveforms for feature extraction.
- Extracting the speech features from the reference and test speech data.
- Computing the variability in the spoken word spoken by the same speaker applying the DTW pattern matching approach.
- Analyzing DTW results using statistical measurements such as probability distribution.

- Developing a speaker-dependent small vocabulary word recognition system using DTW non-linear alignment and a simple decision rule based on DTW distance.
 - Comparing the DTW results with HSM and two speech recognition software packages, Windows XP as Now and Dragon Naturally Speaking (9.0).

Variability such as changes in the speaking rate, speaking style, a speaker speaking the same speech twice exhibit timing differences in the speech sequences. The problem of time alignment is addressed by the DTW algorithm through warping a template or model in an attempt to align key similarities between the test utterance (word) and training templates.

The system was tested in the hospital of the University in Calgary. There one used the weekdays as words and some other additional words. In the testing one used 100 training and 25 test data. First the probability distribution is shown (Table 23.1).

The probability distribution of DTW distance score using 100 training and 25 test samples is: data (mean = 25.59, standard deviation = 2.93, kurtosis = 3.21, skewness = 0.74 for close test and mean = 27.33, standard deviation = 2.66, kurtosis = 4.36, skewness = 0.67) for open test. The probability distribution is shown in Fig. 23.2.

23.2.4 Noise

In all three applications, the speakers are humans and the receivers are machines. It is essential that the machines understand the speech correctly. In a clean environment and using a head set this is today technically possible. This is not anymore the case if noise is present as in all the three examples. The persons should not carry a head set and should not be disturbed by the existence of the speech recognition system. In the examples, we encounter the types introduced earlier. The examples employ most of the methods described in here but will not be repeated.

Table 23.1 Recognition result using 100 training samples and 25 test samples

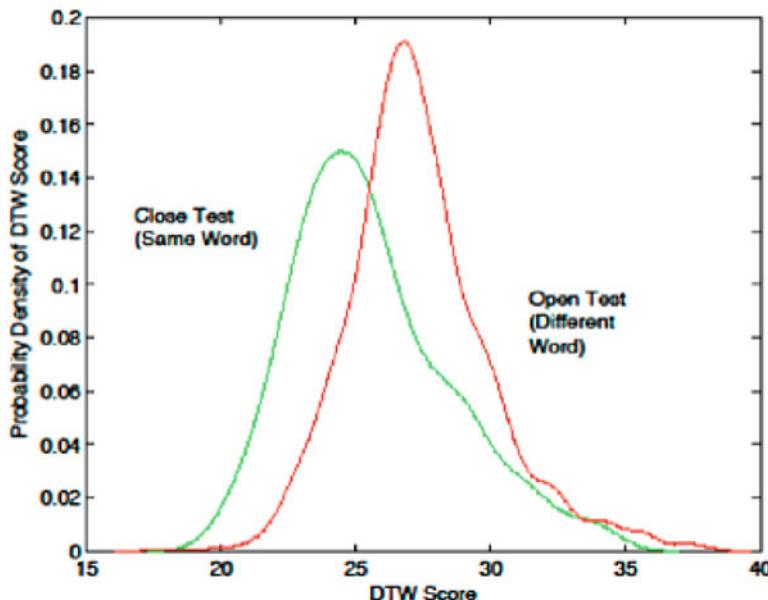


Fig. 23.2 Probability distribution of DTW distance score using 100 training and 25 test data

now. The receiving machines have to perform some actions for which a second machine is necessary. We described the combination of the two machines. It needed some technology that was in principle standard. The acoustic environment in a room depends greatly on the size, shape, and other properties of the confining walls or ceiling. In addition, some external noise can be present as conversations, radio, TV, or other noises. In addition, the speaker may speak in a dialect. The examples are somewhat typical and the situations occur quite often. All types of noise introduced above will occur here. In particular, strong noise can occur if a heavy object is falling down.

23.3 Background Information

General

The three applications in this chapter have a common structure but differ in many aspects. The idea is that a reader will have an own application. If it has a structure that is presented then the structural elements can be taken over. However, many aspects will be different and have to be newly designed. Past A classical toolbox of signal analysis is the (windowed) Fourier transform. In this case, the incoming signal is decomposed into its frequency spectrum, i.e. the original signal, which represents the volume with respect to the time, is transformed to a representation

in terms of the amplitudes of the included frequencies. This information is helpful, if one wants to distinguish different pitches of sounds. In speech signal processing and analysis, a “Localized Trigonometric Transformation (LTT)” is proved a better tool than wavelet analysis. The LTT has similar properties as wavelet does but they are not the same with respect to procedural steps. When analyzing a sound, the original signal represents the sound purely in terms of how the amplitude varies over time, while Fourier analysis represents the sound purely in terms of the frequencies that occur in it. Neither of these representations can tell that a note of a certain frequency was played at a certain time! Wavelets, on the other hand, can provide a “mixed-mode” analysis that contains information about both frequency and time. The windowed Fourier transform subdivides the time interval into small subintervals of equal length and performs the procedure on each subinterval separately. However, this procedure is too inflexible for speech recognition.

Suggestion

It is known that different classes of phonemes need to be characterized in their time-frequency behaviour in different ways. For instance, vowels correspond to relatively long time intervals and can be characterized by a precise measurement of the three dominant (i.e. associated to the highest energy) frequencies whereas plosives can be characterised by the point of time and the length of the time interval. Hence, vowels require a high resolution in the frequency domain and plosives require a high resolution in the time domain. Both cannot be achieved simultaneously in analogy to Heisenberg’s uncertainty principle. Consequently, the windowed Fourier transform with its fixed sizes of the time and the frequency windows appears not to be an ideal choice.

For the speech application we have selected three examples that are motivated by different types and different noise removal techniques. The examples show that a successful application of a speech recognition system requires a close cooperation with the environment. The main target was not the complexity of the system but rather its usefulness. This was also the reason to get financial support.

The three applications have been implemented and have been described in detail in Paul et al. (2013). See also Paul et al. (2012). Details about Health Care Support are in Hebinger et al. (2008).

Commercial systems that have been tried are Dragon (2005) and Microsoft (2006). More implementation aspects can be found in Becheti and Ricotti (2002).

Dynamic features are described in Furui (1986). All the presented examples have been implemented and tested in a realistic environment. The example in the lab was tested in two factories in the cities of Mainz and Kaiserslautern. The example of supporting disabled persons was developed in a project of the state Rheinland-Pfalz in a home for elderly persons in Trippstadt, Germany. The application connected with dementia was developed in the master’s thesis of Paul (2009) at the University of Calgary, Paul (2009) and tested in the University of Calgary.

23.4 Exercises

Exercise 1 (For Medical Experts) Describe another possible medical application in analogy to the one with patients with possible dementia. The goal is to diminish the participation of medical doctors.

Exercise 2 Suppose in your room windows and related objects are repaired. There are several workers and it is somewhat noisy. The workers need always some equipment like thrilling machines. Describe them formally.

Exercise 3 Describe:

Its recognition task and its solution.

The communication with a machine that handles the submission.

For this second machine it has only to be described what it has to do.

Exercise 4 Define a vocabulary for the commands for supporting elderly persons. Define a similarity measure that you can use for clustering.

Exercise 5 Give two similarity measures sim1 and sim2 for the vectors $u = (1, 1, 0)$, $v = (0, 1, 1)$, $w = (1, 0, 1)$ such that $\text{sim1}(u, v) < \text{sim1}(u, w)$ and $\text{sim2}(u, w) < \text{sim2}(u, v)$.

Exercise 6 Give list of actions you want perform on your laptop. Add a list of short but understandable commands for the actions. Describe what one has to do in order to perform the actions corresponding to the commands.

References

- [Paul2014] Paul, S.. Dynamic Automatic Noisy Speech Recognition System (DANSR). Dissertation Kaiserslautern 2015.
- [PaulRichter2013] Paul, S., Richter, M.M., Michel, V. (2013). Reaction to Hybrid Noise in Communication.
- [PaulRichterLiu2012] Paul, S., Richter, M.M., Liu, S. (2012). Hybrid solution to single-channel hybrid noisy speech for an industrial environment. ISSPIT, Ho Chi Minh City.
- [Hebinger2008] Hebinger, G., Michel, V., Richter, M.M., Simon, A. Speech Recognition Support of Assisted Living. Funktionalanalysis und GEOnamatik, Bericht 40. (2008).
- [Dragon2005] Dragon (2005): Naturally Speaking. <http://www.dragonsys.com/>.
- [Microsoft2006] Microsoft (2006) Voice Recognition Software, Say Now. <http://www.say-now.com/>.
- [Furui1986] Furui, S. (1986). Speaker independent isolated word recognition using dynamic features of speech spectrum. IEEE Transactions on Acoustics Speech, and Signal Processing, 34.
- [Becheti2002] Becheti, C. and P. L. Ricotti (2002). Speech Recognition Theory and C++ Implementation. John Wiley and Sons Ltd., Italy, 2002.
- [Pau2009] Sheuli Paul (2009) Dynamic Time Warping for Small Vocabulary Word Recognition. Master Thesis University of Calgary.

Chapter 24

Different Speech Applications—Part B



Overview

Different aspects of machine learning to a few practical examples such as Wake Up Word (WUW), Speech Analysis and Sound Effects Laboratory (SASE Lab), Wake-Up-Word: Tool Demo, and Elevator Simulator are presented in this chapter. The aim is to give an essence of understanding, namely, how humans communicate using spoken language.

24.1 Introduction

Starting with the goal of describing what our aim is, that is, providing brief overview of what constitutes machine intelligence in the context of a language, this loaded question, can be answered by breaking it down into the following:

- Algorithms and Technologies
There a number of technologies and algorithms developed for purpose of aiding language understanding. The non exhaustive list below just provides a glimpse:
 - Regular Expressions and Finite State Automata
 - N-grams
 - Hidden Markov Models and Automatic Speech Recognition
 - Context Free Grammars
- Syntactic and Semantic Analysis of a Language
Ultimately the tools that were developed had as a goal to help human develop the solution that is capable of understanding the human speech.
 - Parsing with Context-Free Grammars
 - Statistical Parsing
 - Semantics: Representing Meaning

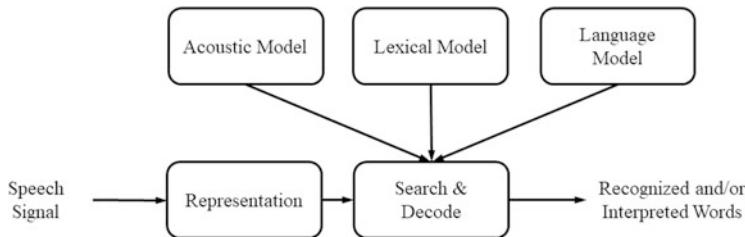


Fig. 24.1 Search and decoding process is depicted in the figure above

- Computational Semantics
- Lexical Semantics
- Computational Lexical Semantics
- Applications
 - The solution that is packaged in form of application:
 - Discourse Management
 - Information Extraction
 - Question - Answering and Summarization
 - Machine Translation

The best way to depict this process is visually (Fig. 24.1):

A stating point in any human communication is to understand how humans produce speech (Fig. 24.2).

We have developed *SASE_LAb* (Speech Analysis and Sound Effects Laboratory), a MATLAB tool to help interested individuals understand how the speech is being produced and more. This MATLAB tool was developed by me, my student, Jacob Zurasky, and with help of NSF grant and J. Rebecca Dowell. The directions how to use are provided in https://fltech-my.sharepoint.com/:w/g/personal/vkepuska_fit_edu/Efkf250Qb-dGkzPWimCQgi8BcxHL_JVNEW-JfB5kJfhJjw?e=eo4gR7.

Speech processing is the application of signal processing techniques to a speech signal for a variety of applications. A common goal between applications is to represent characteristics of the original signal as efficiently as possible. Example applications include speech recognition, speaker recognition, speech coding, speech synthesis, and speech enhancement. Efficiency is a key factor in all of these applications. These applications are covered in more detail in the motivation section of this chapter. In speech recognition, the goal is to provide a sound wave containing speech to a system, and have the system recognize parts of words or whole words. To complete this goal, speech processing must first take place to prepare the speech signal for recognition. This is done by breaking down the speech signal into frames of approximately 10–30 milliseconds, and generating a feature vector that accurately and efficiently characterizes the speech signal for that frame of time. Reducing the frame from a fairly large set of speech signal samples to a

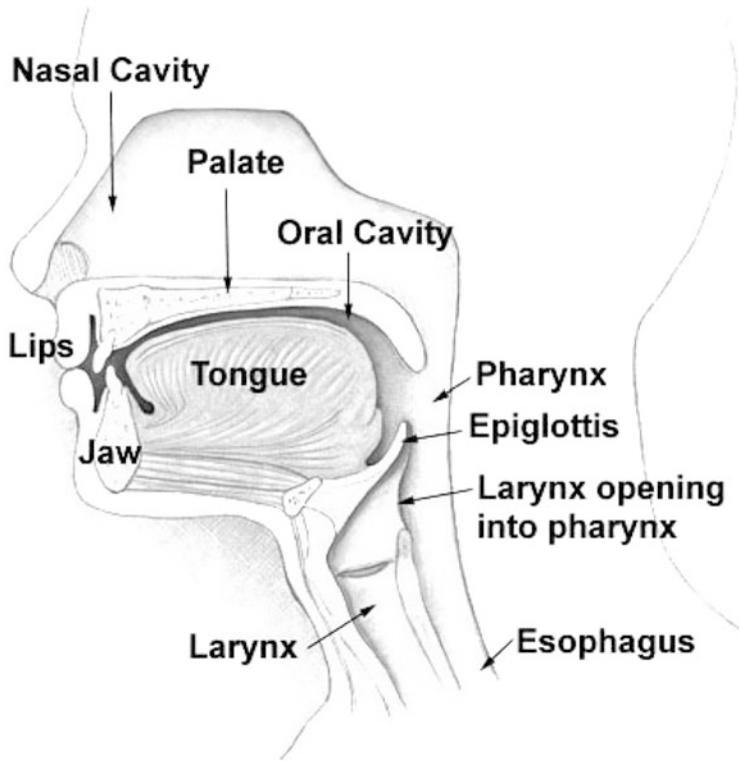


Fig. 24.2 Figure depicting human anatomy and the organs involved in speech production. <http://training.seer.cancer.gov/head-neck/anatomy/overview.html>,<https://commons.wikimedia.org/w/index.php?curid=1678037>



Fig. 24.3 Figure depicting block diagram of speech recognition system

much smaller set of data, the feature vector, allows for quicker computation during the recognition stage of the system. Figure 24.3 shows a high level diagram of a typical speech recognition system. The input to the system is a speech signal and is passed to the front end of the system. The front end is responsible for the speech processing step and will extract features of the incoming speech signal. The back end of the system will use the features provided by the front end and based on statistical models, provide recognized speech. Our discussion will first focus on the front-end and related speech processing required to extract feature vectors from a speech signal. Speech Processing is an important technology that is used widely by many people on a day-to-day basis. All smartphones come equipped with speech recognition capabilities to enable hands free use of certain phone functionality.

A recent advancement of this mobile technology is Siri on the Apple iPhone. This application takes speech recognition a step further and adds machine understanding of the user's requests. Users can ask Siri to send messages, make schedules, place phone calls, etc. Siri responds to these requests in a human-like nature, making the interaction seem like almost talking to a personal assistant. At the base of this technology, speech processing is required to extract information from the speech signal to allow for recognition and further more understanding.

Speech recognition is also commonly used in interactive voice response (IVR) systems. These systems are used to handle large call volumes in areas such as banking and credit card services. IVR systems allow interaction between the caller and the company's computer systems directly by voice. This allows for a large reduction in operating costs, as a human phone operator is not necessary to handle simple requests by a customer. Another benefit of an IVR system is to segment calls to a large company based on the caller's needs and route them to appropriate departments. Other applications of speech processing and recognition focus on a hands free interface to computers. These types of applications include voice transcription or dictation systems. These can be found commercially in use for direct speech to text transcription of documents. Other hands-free interfaces allow for safer interaction between human and machines such as the OnStar system used in Chevy, Buick, GMC, and Cadillac vehicles. This system allows the user to use their voice to control navigation instructions, vehicle diagnostics, and phone conversations. Ford vehicles use a similar system called Sync, which relies on speech recognition for hands free interface to calling, navigation, in-vehicle entertainment, and climate control. These systems use of hands free interface to computing, allows for a safer interaction when the users attention needs to be focused on the task at hand, driving.

24.2 Discrete-Time Signals

A good understanding of discrete-time signals is required prior to discussing the foundations of speech processing. Computers are discrete systems with finite resources such as memory. Sound is stored as discrete-time signals in digital systems. The discrete-time signal is a sequence of numbers that represent the amplitude of the original sound before being converted to a digital signal. Sound travels through the air as a continuously varying pressure wave. A microphone converts an acoustic pressure signal into an electrical signal. This analog electrical signal is a continuous-time signal that needs to be discretized into a sequence of samples representing the analog waveform. This is accomplished through the process of analog to digital conversion. The signal is sampled a rate called the sampling frequency (F_s) or sampling rate. This number determines how many samples per second are used during the conversion process. The samples are evenly spaced in time, and represent the amplitude of the signal at that particular time. The above figure shows the difference between a continuous-time signal and a discrete-time signal. On the left, one period of a 200 Hz sine wave is shown. The period

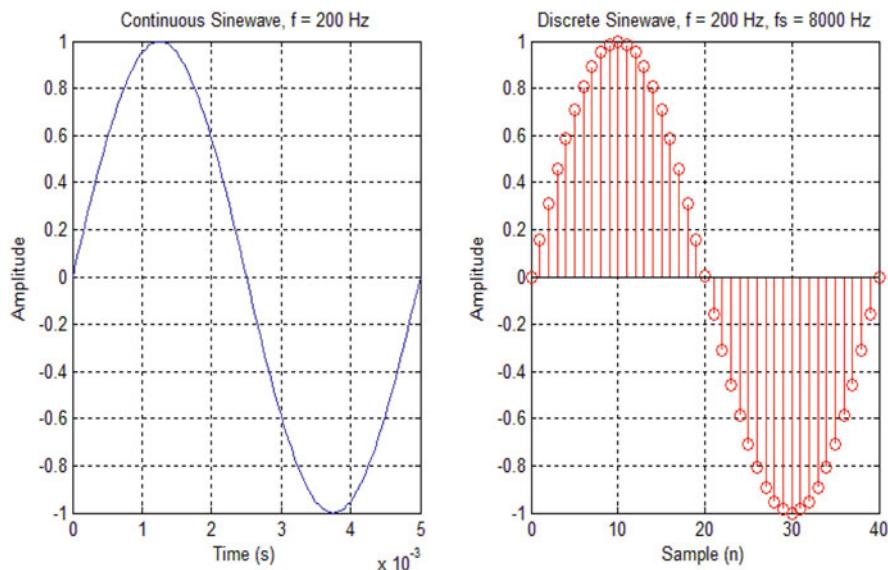


Fig. 24.4 Continuous and discrete time signal

of this signal is the reciprocal of the frequency and in this case, five milliseconds. On the right, the signal is shown in discrete-time representation. The signal is a sequence of samples, each sample representing the amplitude of the signal at a discrete time. The sampling frequency for this example was 8 kHz, meaning 8000 samples per second. The result of one period of the 200 Hz sine wave is 40 samples (Fig. 24.4).

The sampling frequency is directly related to the accuracy of representation of the original signal. By decreasing the sampling rate to 2 kHz, or 2000 samples per second, the discrete-time signal loses accuracy. This can be seen on the right side of the following figure (Fig. 24.5).

The exact opposite is true by increasing the sampling frequency of the signal. A discrete-time signal can be represented more accurately. The following figure uses a sampling frequency of 44.1 kHz. It can be seen that the signal on the right more accurately describes the continuous-time signal at a sampling rate of 44.1 kHz as opposed to 2 kHz or 8 kHz (Fig. 24.6).

Common sampling rates used currently for digital media are as follows:

- 8 kHz—Standard land-line telephone service
- 16 kHz—Wideband land-line and cellular telephone service
- 44.1 kHz—CD Audio Tracks
- 48 kHz—DVD Audio Tracks

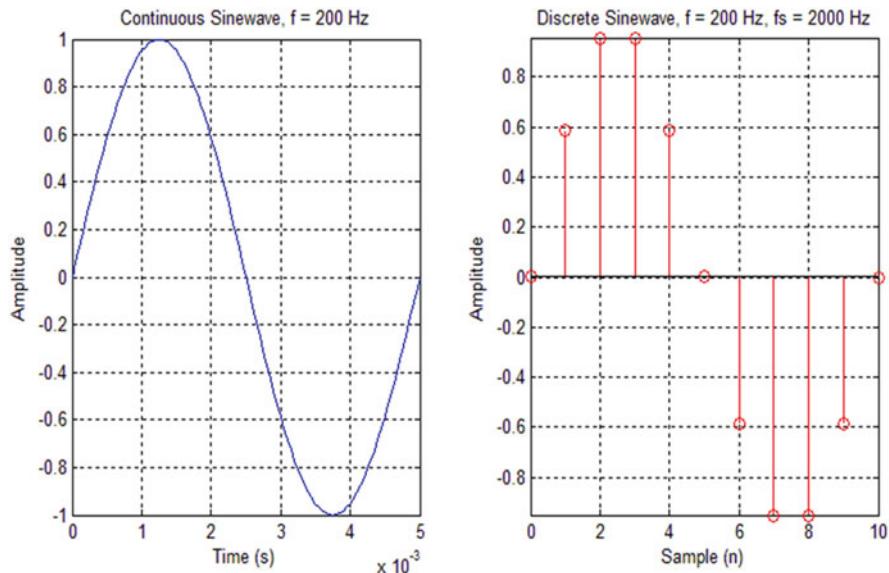


Fig. 24.5 Continuous and discrete time signal with reduced sampling rate

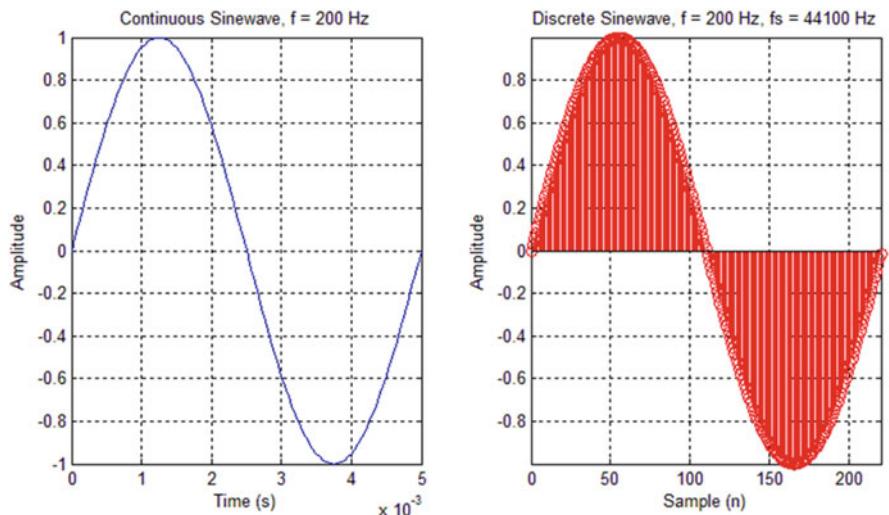


Fig. 24.6 Continuous and discrete time signal sampled at the higher rate of 44.1 kHz

24.3 Speech Processing

The main goal of speech processing is to reduce the amount of data used to characterize the speech signal while maintaining an accurate representation of the

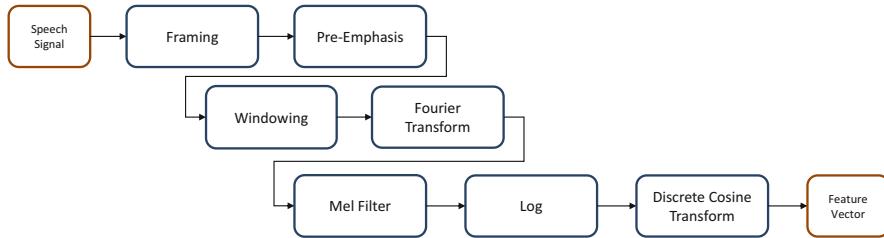


Fig. 24.7 Front-end stage of a typical Mel-filter Cepstral Coefficients based speech recognition system

original data. This process produces a feature vector to typically 13 numbers. The feature vector is commonly referred to as *Mel-Frequency Cepstral Coefficients (MFCCs)*. The process of feature extraction can be broken down into several stages of mathematical operations that take place on a discrete-time signal input. The following is high level diagram of feature extraction stages (Fig. 24.7).

24.3.1 *Framing*

The speech signal can be of any length, but for analysis, the signal must be divided in segments. Each segment, or frame, will be analyzed and a feature vector will be produced. Speech signals are typically stationary over a period of 10–30 milliseconds. Given a sampling frequency of 8 kHz, corresponding frame sizes are of 80 to 256 *samples*. These samples contained in the frame will be passed through all stages of the front end to produce a vector containing typically 13 values that characterize the speech signal during that frame.

Upon complete processing of a particular frame, the next frame should not begin where the previous one ended. To more accurately process the signal, the next frame should overlap the previous frame by some amount.

In the presented Fig. 24.8 shows 768 samples of a speech signal and also the overlapping nature of the speech frames. The blue signal is the speech signal and it can be noted this is a semi-stationary section of speech. The periodicity of the signal is clearly shown. The other signals show how this speech would be divided into five frames. Each colored curve is an analysis window that segments the speech signal into frames. Each frame is 256 samples in length, and each frame overlaps the previous by 50%, or 128 samples in this case. This insures accurate processing of the speech signal. A front end system can be described by its frame rate, or the number of frames per second that the speech signal is divided into. The frame rate of the front end also translates into the number of feature vectors produced per second due to the fact that one frame produces one feature vector.

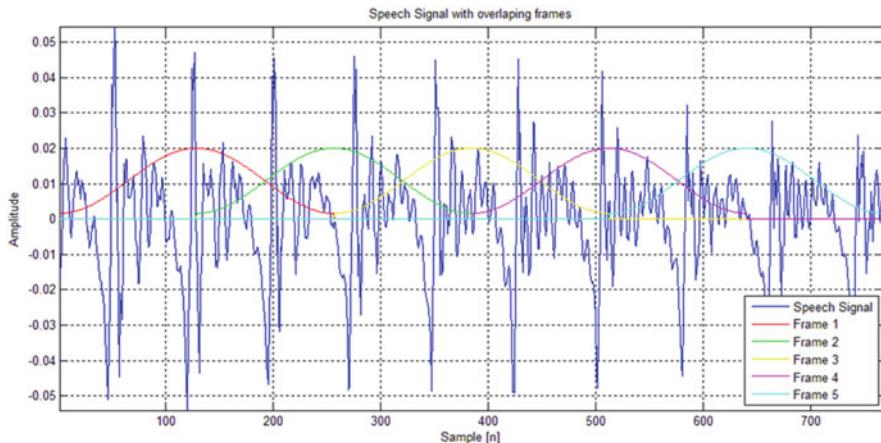


Fig. 24.8 Speech signal with overlapping frames

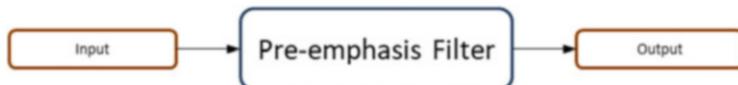


Fig. 24.9 Pre-emphasis filter block diagram

24.3.2 *Pre-emphasis*

The next stage of the front end is to apply a pre-emphasis filter to the speech frame that has been segmented in the previous step. A pre-emphasis filter in relation to speech processing is typically a high-pass, 1st order, finite impulse response (*FIR*) filter. A filter modifies a signal that is passed through it (Fig. 24.9).

A filter has a characteristic called frequency response. This describes how the filter modifies the signal passed through it. The filter used here is a high-pass, meaning that it will pass the frequencies above the cut-off frequency, while attenuating or reducing parts of the signal below the cut-off frequency. The frequency response of a common 1st order pre-emphasis filter is shown below (Fig. 24.10).

In the provided figure, the used parameter was of the value of 0.97 for the filter coefficient and the sampling rate of 8 kHz. The frequency response of this filter shows that the magnitude of lower frequencies are attenuated or reduced in magnitude. The opposite is also true for higher frequencies. The reason for applying the pre-emphasis filter is tied to the characteristics of the human vocal tract. There is a roll-off in spectral energy towards higher frequencies in human speech production. To compensate for this factor, lower frequencies are reduced. This prevents the spectrum from being overpowered by the higher energy present in the lower part of the spectrum (Fig. 24.11).

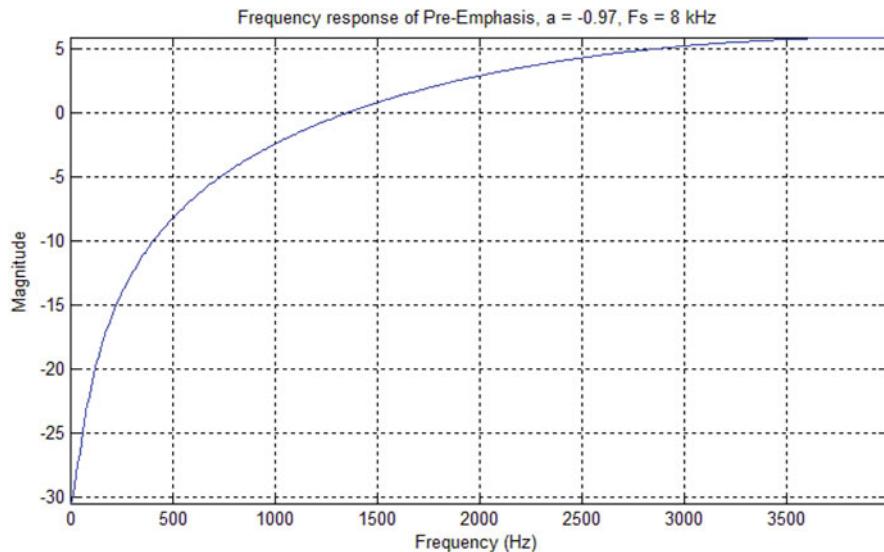


Fig. 24.10 Pre-emphasis filter frequency response

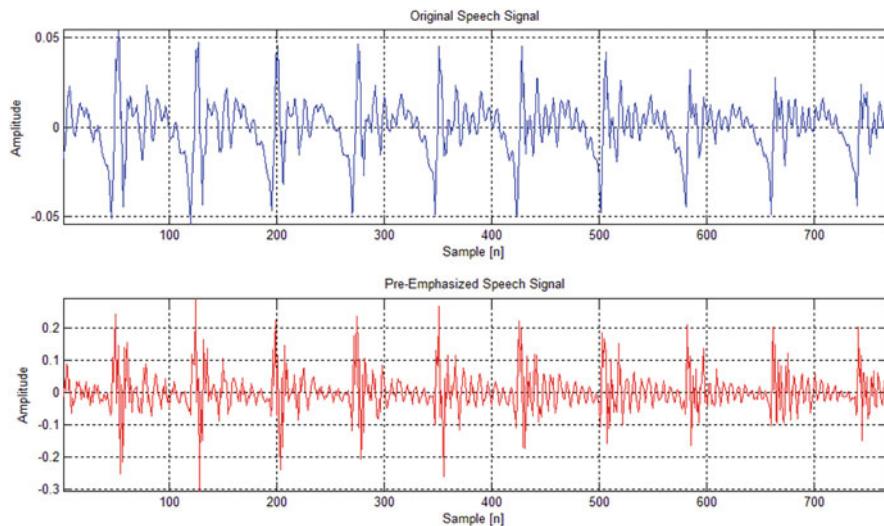


Fig. 24.11 Original and pre-emphasised signal

The above figure shows the original speech signal in blue and the pre-emphasized speech signal in red. While maintaining the same overall periodicity and general waveform shape, the high frequency components are accentuated. The quicker changing parts of the signal, higher frequencies, are compensated so that the lower frequency energy does not overpower the spectral results. The operation of applying

a filter to a signal is represented mathematically through the convolution operation, denoted by the ‘*’ operator depicted below.

$$y(t) = x(t) * h(t)$$

The above equation convolves the input signal $x(t)$ with filter $h(t)$ to produce the output $y(t)$. In a continuous time signal, the convolution operation is defined by the following integration:

$$y(t) = \int_{-\infty}^t h(\tau)x(t - \tau)d\tau$$

In a discrete-time system the convolution operation changes from integration to summation:

$$y(t) = \sum_{i=0}^N \beta_i x[n - i]$$

where:

- N —filter order
- β_i —filter coefficients
- $x[n]$ —input signal
- $y[n]$ —output signal

In the case of our pre-emphasis filter, the order is one. This means there will be two coefficients, β_0 and β_1 . The first coefficient of a *FIR* filter, β_0 , is always one. The coefficient β_1 used in the above example frequency response with the value -0.97 . Expanding the above summation based on these coefficient values yields the following results.

$$y[n] = \beta_0 x[n] + \beta_1 x[n - i]$$

$$y[n] = x[n] - 0.97x[n - i]$$

The input to this function is the sequence of samples in the speech frame, denoted here by $x[n]$. The output of this filter is the pre-emphasized (high-pass filtered) frame of speech, denoted $y[n]$.

24.3.3 Windowing

After a frame of speech has been pre-emphasized, a window function must be applied to the speech frame. While many different types of windowing functions

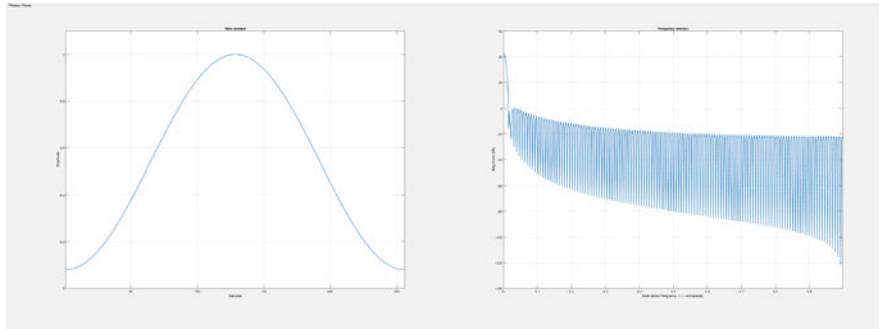


Fig. 24.12 Hamming Window of length 256 and its frequency characteristics depicted to the right

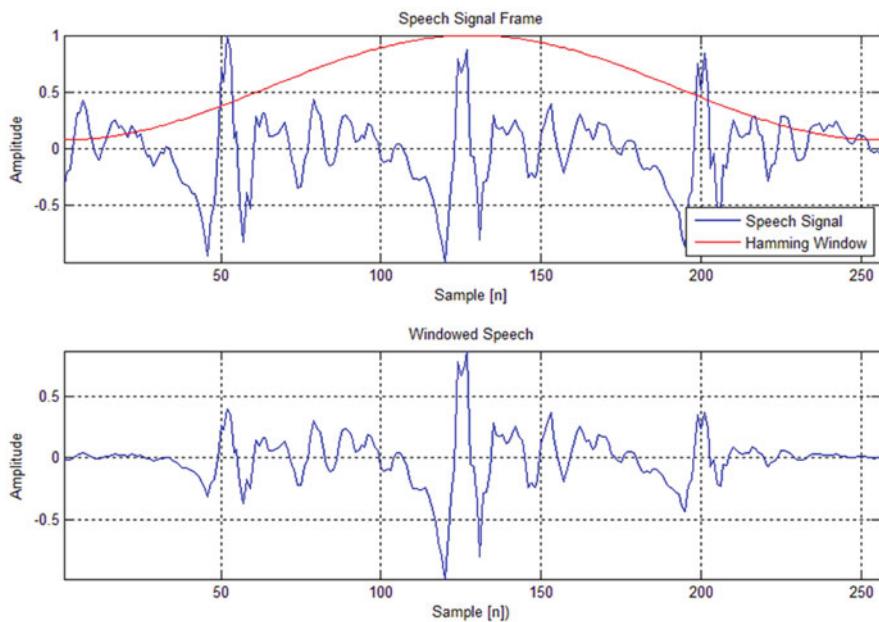


Fig. 24.13 Figure depicting: (a) Speech signal with overlaid Hamming Window. (b) Speech signal after the windowing operation

exist (Chaps. 11 and 2), Chebyshev, Gaussian, Hanning, Keiser, Tukey to name a few, a Hamming window is typically used for speech processing. The figure below shows a hamming window of length 256 (Figs. 24.12, and 24.13).

The general equation is provided below where it is assumed the filter is of any length N .

$$w[n] = 0.54 - 0.46 \cos \left(\frac{2\pi n}{N-1} \right)$$

To apply this window function to the speech signal, each speech sample is multiplied by the corresponding value of the window function to generate the windowed speech frame. At the center of the hamming window, the amplitude is 1.0 and decays to the value 0.08 at either the beginning or end of the hamming window. This allows for the center of the speech frame to remain relatively unmodified by the window function, while samples are attenuated more, the further they are from the center of the speech frame. Observe the following figure. On the top half, a speech signal is shown in blue, with a hamming window function in red. The bottom half of the figure shows the results when the hamming window is applied to the speech signal. At the center of the frame, the speech signal is nearly the original values, but the signal approaches zero at the edges.

This process of windowing is very important to speech processing. In the next stage, the Fourier Transform, we apply the windowed speech frame data. If a windowing function is not applied to a speech frame, there can be large discontinuities at the edges of the frame. These discontinuities will cause problems with the Fourier Transform and will induce errors in the frequency spectrum of the framed audio signal. While it may seem like information is being lost at the edges of the speech frame due to the reduction in amplitude, the overlapping nature of sequential speech frames ensures all parts of the signal are analyzed.

24.3.4 Fourier Transform

The Fourier Transform is an algorithm used to transform a time domain signal into a frequency domain. While time domain gives information about how the signal's amplitude changes in time, frequency domain shows the signals energy content at different frequencies. See the following graph for an example frequency spectrum of a time domain signal. The x-axis is frequency and the y-axis is magnitude of the signal. It can be observed that this particular frequency spectrum show a concentration of energy below 1 kHz, and another peak of energy between 2.5 kHz and 3.5 kHz (Fig. 24.14).

The human ear interprets sound based on the frequency content. Speech signals contain different frequency content based on the sound that is being produced. Speech processing systems analyze frequency content of a signal to recognize speech. Every frame of speech passed through the speech processing system will have the Fourier Transform applied to allow analysis in frequency domain.

The above graph shows peaks of frequency magnitude in three areas. Most speech sounds are characterized by three frequencies called *formant* frequencies. The formants for a particular sound are resonant frequencies of the vocal tract during that sound and contain the majority of signal energy. Analysis of formant locations in terms of frequency is the basis for recognizing particular sounds in speech.

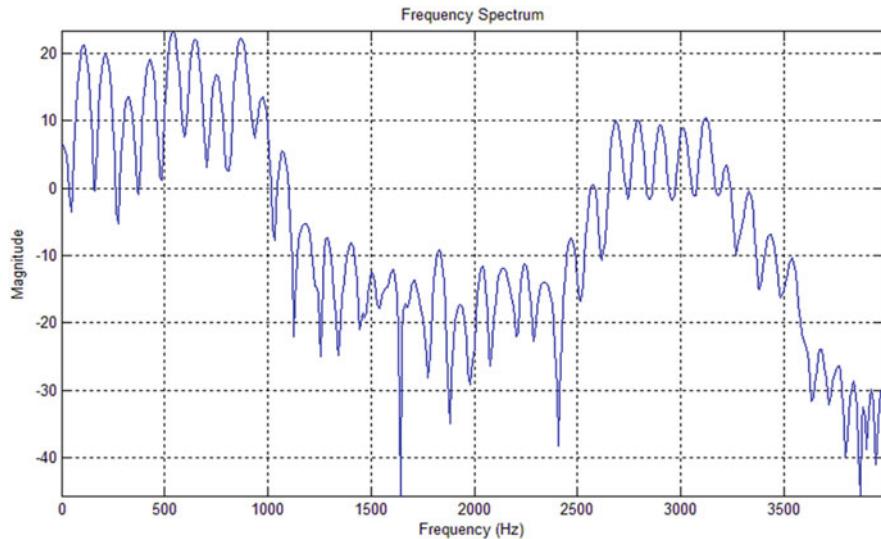


Fig. 24.14 Spectrum of a speech frame depicting its content in frequency

The Fourier Transform $\hat{f}(\omega)$ of continuous time signal $f(x)$ is defined as:

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-2\pi jx\omega} dx, \quad \text{for every real number } \omega$$

The Inverse Fourier Transform of $\hat{f}(\omega)$ reproduces the original signal $f(x)$:

$$f(x) = \int_{-\infty}^{+\infty} \hat{f}(\omega)e^{-2\pi jx\omega} d\omega, \quad \text{for every real number } x$$

where: $\hat{f}(\omega)$ —continuous frequency spectrum, $f(x)$ —continuous time signal.

Speech processing typically deals with discrete-time signals, and the corresponding discrete Fourier Transforms pair are given below:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi \frac{k}{N}n}$$

and

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{j2\pi \frac{k}{N}n}$$

where

- $x[n]$ —discrete time signal
- $X[k]$ —discrete frequency spectrum
- N —Fourier Transform size
- n —sample number
- k —frequency bin number

The vector $X[k]$ contains the output values of the Fourier Transform algorithm. These values are the frequency domain representation of the input time domain signal, $x[n]$. For each index of k , from 0 to N , the value of the vector is the magnitude of signal energy at the frequency bin k . When analyzing magnitude, the Fourier Transform returns results that are symmetric across the mid-point of the FFT size.

Due to this symmetry, only the first half of the Fourier Transform is used when analyzing the magnitude of frequency content of a signal. The relation from k to actual frequency depends on the sampling rate (F_s) of the system. The first frequency bin, $k = 0$, represents 0 Hz, or the overall energy of the signal. The last frequency bin, $k = N/2$, represents the maximum frequency that can be detected based on the sampling rate of the system. The Nyquist-Shannon Sampling Theorem states the maximum frequency that can be detected in a discrete-time system is half of the sampling rate. Given a sampling rate of 8 kHz, the maximum frequency, or Nyquist Frequency, would be 4 kHz.

Each frequency bin k , represents a range of frequencies rather than a single value. The range covered by a set of frequencies is called a bandwidth (BW). The bandwidth is defined by initial and final frequencies in the given band. For example if a frequency bin started at 200 Hz and ended at 300 Hz, the bandwidth of the bin would be 100 Hz. The Fourier Transform returns frequency bins that are equally spaced from 0 Hz to the Nyquist Frequency, with each bin having the same bandwidth. To compute the bandwidth of each bin, the overall bandwidth of the signal must be divided by the number of frequency bins. For example given $F_s = 8$ kHz and $N = 1024$:

$$BW_{bin} = \frac{\left(\frac{F_s}{2}\right)}{\left(\frac{N}{2}\right)} = \frac{\left(\frac{8\text{kHz}}{2}\right)}{\left(\frac{1024}{2}\right)} = 7.8125\text{Hz}$$

The computed bandwidth provides a measure of sensitivity of applied FFT, that is, provides a *bandwidth per bin*. To translate from frequency bin k to actual frequency, the bin number (k) is multiplied by the bin bandwidth (BW_{bin}). For example if $BW_{bin} = 7.8125$ and $k = 256$ we could compute the following:

$$f_{max_{bin}} = BW \times k = 7.8125 \frac{\text{Hz}}{\text{bin}} \times 256 = 2000 \text{ Hz}$$

$$f_{min_{bin}} = f_{max_{bin}} - BW = 2000\text{Hz} - 7.8125\text{Hz} = 1992.1875\text{Hz}$$

These calculations show that frequency bin 256 covers frequencies from about 1.992 kHz to 2 kHz. Note the equation for bin bandwidth is indirectly proportional to the Fourier Transform size. As the Fourier Transform size increases, the bin bandwidth decreases, thus allowing a finer resolution in terms of frequency. A finer resolution in frequency produces more accurate results in terms of the original frequency content of the signal versus the output from the Fourier Transform.

An optimization of the Fourier Transform is the Fast Fourier Transform (*FFT*). This is a much more efficient way to compute the Fourier Transform of a given signal. There are many different algorithms for computing the *FFT* such as the Cooley-Tukey algorithm. Many algorithms rely on the divide-and-conquer approach, where the overall Fourier Transform is computed by breaking down the computation into smaller Fourier Transforms. Direct implementation of the Fourier Transform is of order N^2 while the Fast Fourier Transform achieve a much lower order of $N \log(N)$. Another optimization used is pre-computing 'twiddle' factors. The complex exponential from Fourier Transform definition is known as the twiddle factor. The value of the complex exponential is independent of the input signal $x[n]$ and is always the same value for a given n, k and N . Since these values never change for a particular n and k , a table of values of size n by k can be computed ahead of time. This look-up table has every possible value of the complex exponential for a given n and k . Rather than computing the exponential every time, the algorithm 'looks up' the value in the table. This greatly improves the efficiency of the Fourier Transform algorithm.

24.3.5 Mel-Filtering

The next stage of speech processing is converting the output of the Fourier Transform to mel-scale rather than linear frequency. The mel-scale was first introduced in 1937 by Stevens, Volkman, and Newman. The mel-scale is based on the fact that human hearing responds to changes in frequency logarithmically rather than linearly. The frequency of 1 kHz was used as a reference point where 1000 Hz is equal to 1000 mels. The equation relating frequency to mels is as follows (Fig. 24.15):

$$m = 2595 \log_{10} \left(1 + \frac{f}{700} \right)$$

The above graph shows the transformation function from linear scale to logarithmic scale frequency. A greater change in linear frequency is required for the same increment in mel-scale as linear frequency increases. To apply this transformation to the frequency spectrum output of the Fourier Transform stage of speech processing, a series of triangular filters must be created as originally proposed. Each filter will be applied to the linear frequency spectrum to generate the mel-scale frequency spectrum. The number of mel-filters is dependent on the application of the speech

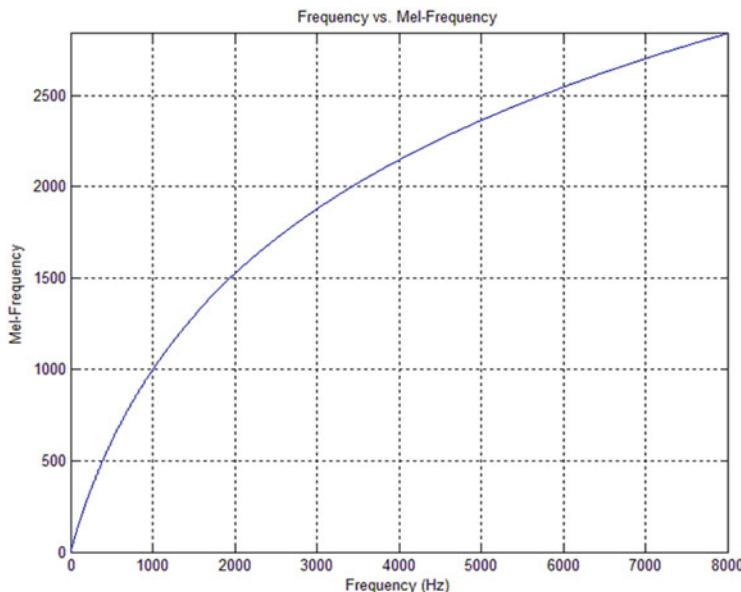


Fig. 24.15 Mel-frequency transformation curve

processing, but typically 20–40 channels are used. The graph below shows 25 mel-filters to be applied to the frequency spectrum obtained from previous section. It can be observed that for each increasing filter, the bandwidth increases, covering a larger range of frequencies. The magnitude of each filter also decreases. This is due to normalization of magnitude according to the bandwidth that the filter covers (Fig. 24.16).

Applying the process of mel-filtering to the frequency spectrum will result in a vector that is the same length as the number of mel-filters that are applied. Each mel-filter function will be multiplied to each value of the frequency spectrum and the results summed. This summation of multiplications will produce a single value corresponding to the magnitude of signal energy at a particular mel-frequency. This process is repeated for each mel-filter.

Every filter channel has a magnitude of zero for all values that fall outside of the triangle, thus eliminating all frequency information related to that mel-filter channel that falls outside of the triangle. Frequencies that are nearest to the center of the mel-filter will have the most impact on the output value, with linearly decreasing significance approach either side of the triangle. See the below figure to observe the input linear frequency spectrum and resulting mel-scale frequency spectrum (Fig. 24.17).

The first graph shows the frequency spectrum obtained from the previous section, the Fourier Transform. The graph below it shows the output after converting to mel-frequency scale. It can be observed that the mel-frequency spectrum has the

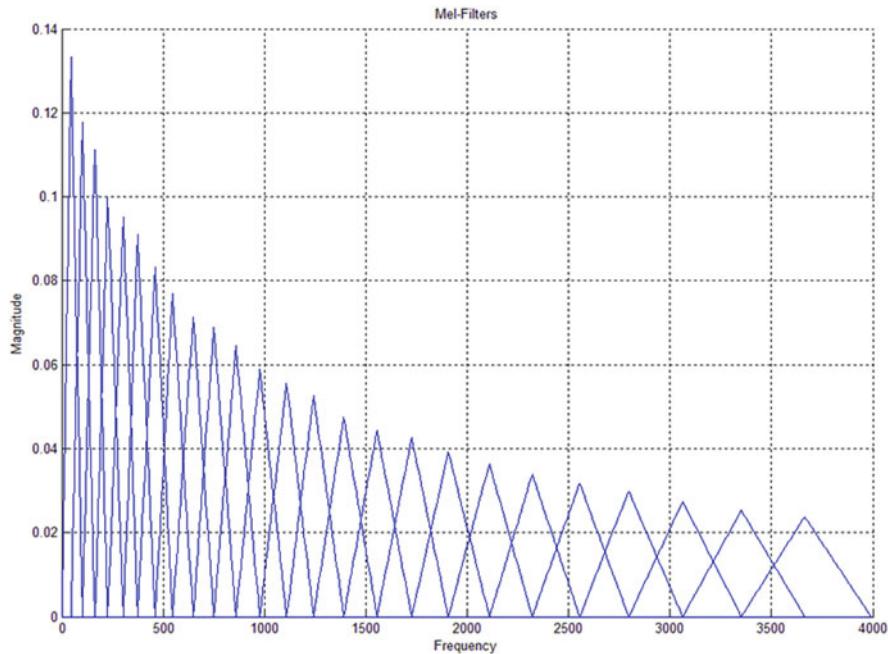


Fig. 24.16 Normalized triangular shaped filters following Mel-scale

same overall shape as the linear scaled frequency spectrum, but the higher frequency information has been compressed.

24.3.6 Mel-Frequency Cepstral Coefficients

The final two steps of speech processing produce results that are called mel-frequency cepstral coefficients or MFCCs. These coefficients form the feature vector that is used to represent the frame of speech being analyzed or processed. As mentioned before, the feature vector needs to accurately characterize the input. The two mathematical processes that need to be applied after the previous steps are taking the logarithm and applying the discrete cosine transform (Fig. 24.18).

The above figure shows the associated feature vector when using the mel-frequency spectrum obtained in the previous section. The first step is to take the logarithm (base 10) of each value in the mel-frequency spectrum. This is a very useful operation, as it allows the separation of signals combined through convolution. For example, if a speech signal is convolved with a noise signal such as background noise:

$$y(t) = x(t) * n(t)$$

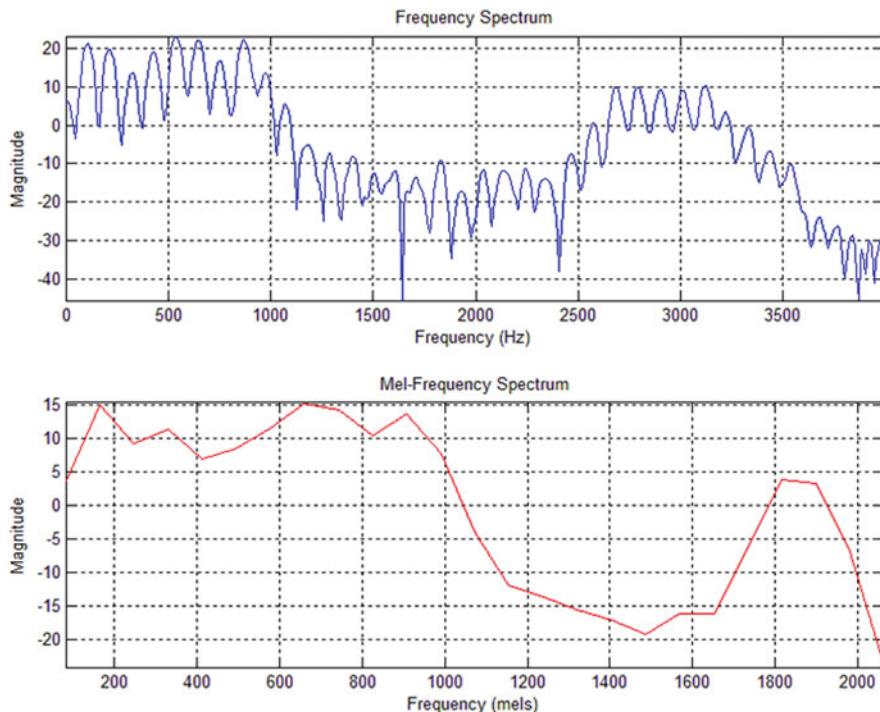


Fig. 24.17 Linear scale vs Mel scale spectrum

where :

$y(t)$ = **Combined speech and noise signal**,

$x(t)$ = **Original speech signal**,

$n(t)$ = **Noise signal**

By taking the Fourier Transform of both sides of the equation, the convolution operation becomes a multiplication. This is due to the convolution property of the Fourier Transform presented below.

$$y(t) = x(t) * n(t)$$

$$Y(\omega) = X(\omega) \cdot N(\omega)$$

Then, by applying the logarithm property of multiplication, the original signal and the noise signal are mathematically added together instead of multiplied. This allows the subtraction of an undesired signal that has been convolved with a desired signal.

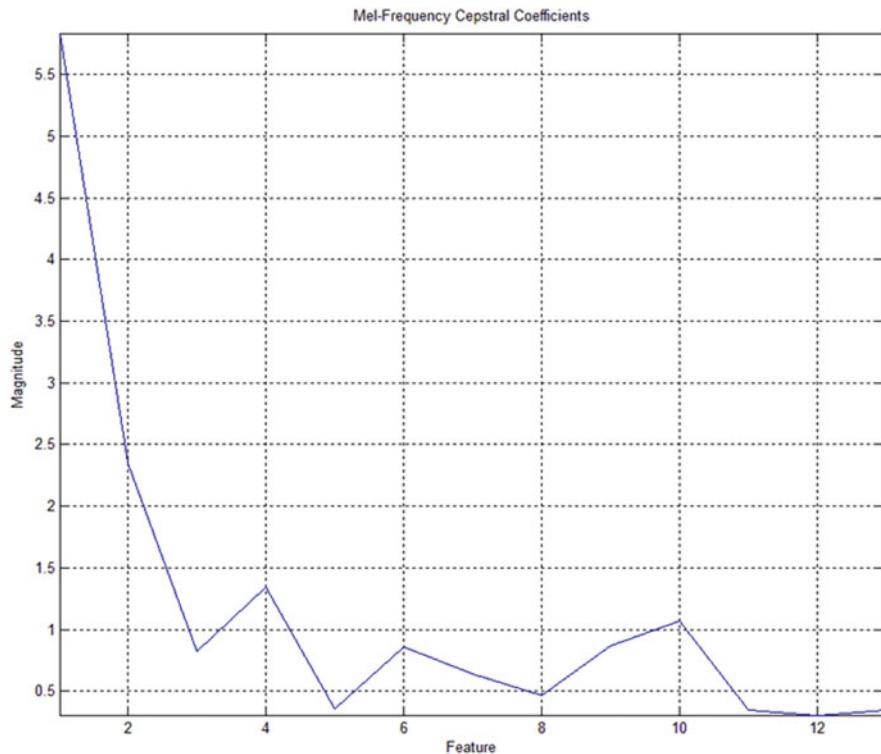


Fig. 24.18 Example of a mel-frequency Cepstral Coefficients of order 12 of a frame of speech

$$Y(\omega) = X(\omega) \cdot N(\omega)$$

$$\log_{10} (Y(\omega)) = \log_{10} (X(\omega)) + \log_{10} (N(\omega))$$

From the last equation, if the noise signal is known, then it can be subtracted from the combined signal. After the logarithm has been taken of each value from the mel-frequency spectrum, the final stage of speech processing is to apply the discrete cosine transformation.

Fourier Transform:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n}$$

$$e^{-j2\pi \frac{k}{N} n} = \cos \left(-2\pi \frac{k}{N} n \right) + j \sin \left(-2\pi \frac{k}{N} n \right)$$

Then by dropping the imaginary component the kernel becomes:

$$\cos\left(-2\pi\frac{k}{n}n\right)$$

The resulting discrete cosine transform equation is:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cos\left(-2\pi\frac{k}{n}n\right)$$

This operation will result in a vector of values that have been transformed from mel-frequency domain to (real) cepstral domain (know as discrete cosine transformation or DCT). This transformation led to the name Mel-Frequency Cepstral Coefficients, MFCCs. Most applications use only the first 13 values to form the feature vector, truncating the remaining results. The length of the feature is dependent on the application, but 13 values is sufficient for most speech recognition tasks.

24.4 Speech Analysis and Sound Effects Laboratory (SASE_Lab)

For this project, a speech processing tool was created with MATLAB to allow analysis of all the steps involved. The tool has been named *SASE_Lab*, for Speech Analysis and Sound Effects Lab. The result is a graphical user interface that allows the user to either record a signal or open an existing waveform for analysis. This signal can then be played back to hear what speech produced. The interface has six plots that show the speech signal at the various stages of speech processing. Starting at the top-left, a plot shows the entire waveform of the original signal. After having opened or recorded a signal, the user can then click on a part of the signal shown to analyze that particular frame of speech. After a frame has been selected, the other five plots show information related to that frame of speech. The top-right plot shows only the frame of speech selected, rather than the whole signal. It also shows the windowing function overplayed in red. On the middle row, left side, the plot shows the frame of speech after the windowing function has been applied. To the right of that plot, the frequency spectrum of the speech frame is shown. On the bottom row, left side, the plot shows the results of converting the linear scale frequency spectrum to mel-scale frequency. The plot on the bottom-right shows the final output of the speech processing, a vector of 13 features representing the input for the particular frame being analyzed (Fig. 24.19).

The above picture is a screenshot of the SASE Lab tool analyzing a particular frame of speech from a signal. The part of the original signal being analyzed is highlighted with a vertical red bar in the top-left plot.

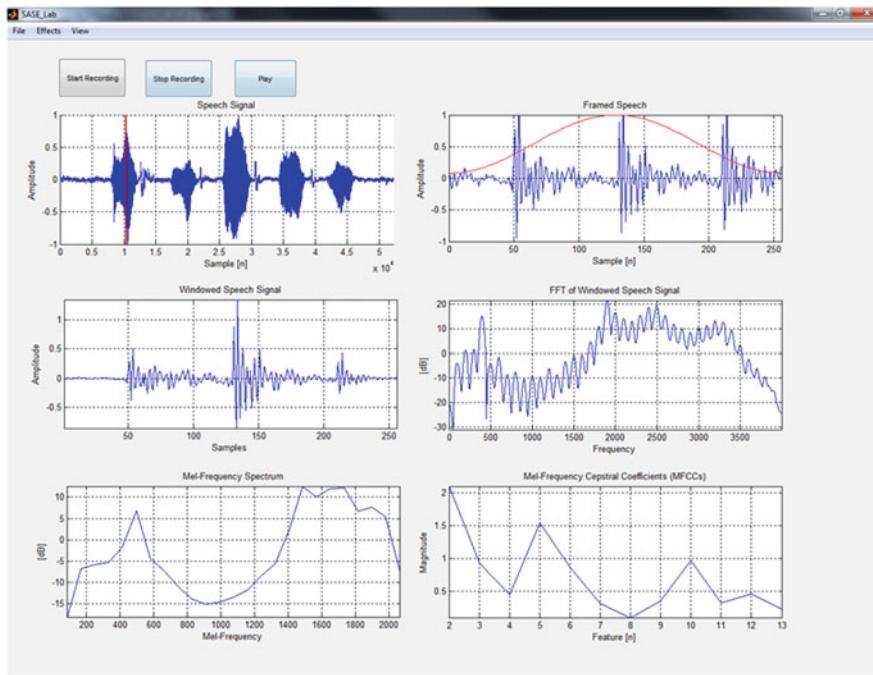


Fig. 24.19 Image depicting the interfacing provided by the MATLAB tool SASE_Lab “Speech Analysis and Special Effects Laboratory”

In addition to these six plots, there are three buttons above them. These three buttons control starting a recording from a microphone, stopping a recording, and playing a recording. Along the menu bar, the user has standard options such as File, View, and Effects. The File menu allows a user to open or save a recording. The View menu allows the user to switch between the view shown above or the spectrogram view. The Effects menu allows the user to apply several different audio effects to the speech signal.

The other main view of the *SASE_Lab* shows spectrogram of the signal. This displays how the frequency content of the signal changes over time. This is an important piece of information when dealing with speech particularly for speech recognition.

The human ear differentiates sounds based on frequency content, so it is important to analyze speech signals for their frequency content (Fig. 24.20).

The screenshot above shows the spectrogram view of SASE Lab. On the top half of the display, the waveform of the original signal is displayed. On the bottom section, the spectrogram of the signal is displayed. The x-axis is time and the y-axis is frequency. The above screenshot is analyzing the speaker saying “a, e, i, o, u”. The frequency content of each utterance shows distinct patterns with regards to frequency content. The spectrogram view of SASE Lab also allows the user

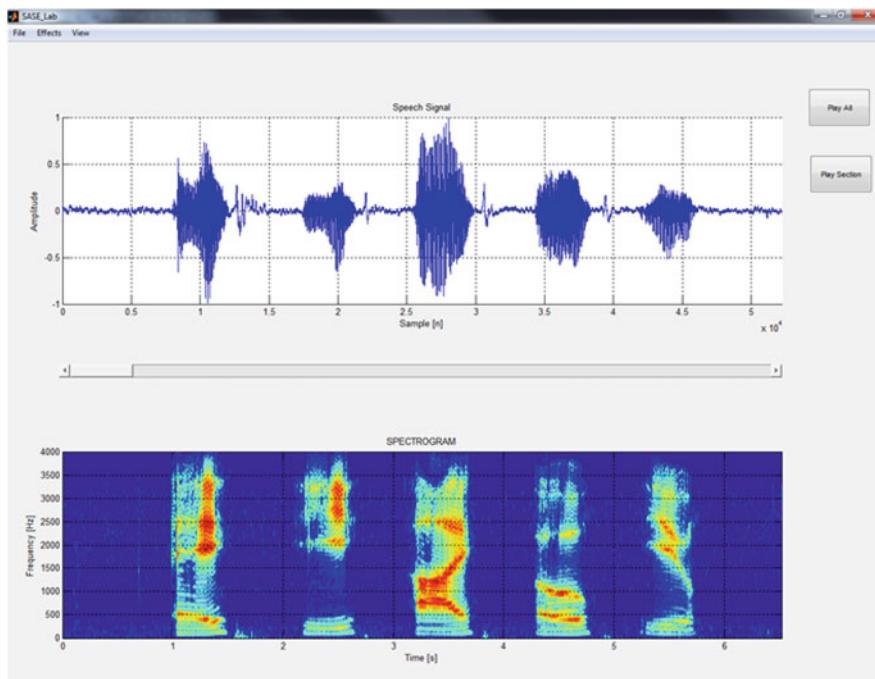


Fig. 24.20 Waveform and Spectrograph analysis performed on the data

to select a 3 second window to analyze separately in the case that a long speech signal is present. When looking at a longer speech signal, the spectrogram become crowded due to compressing a large amount of data in the same display space. See the following screenshot for an example (Fig. 24.21).

The speech signal shown is approximately 18 seconds long and hard to analyze when showing the spectrogram of the whole signal at once. To alleviate this issue, a slider bar was implemented to allow the user to select a 3 second window of the entire speech signal. The window is shown in the top graph by the highlighted red section of the signal waveform. In a new window, the 3 second section of speech waveform is plotted, along with the corresponding section of spectrogram, as depicted from the screenshot below (Fig. 24.22).

This figure shows only the section of speech highlighted in the previous figure. Spectral characteristics are much easier to observe and interpret compared to viewing the entire speech signal spectrogram at the same time. The user can change the position on the slider bar from the main view, and the secondary view will update its content to show the new 3 second selection. The user may also play only the 3 second selection by pressing the Play Section button on the right side of the main view. The Play All button will play the entire signal.

Analysis on the feature vector produced for different vowel sounds yields results as expected. The feature vector should be able to accurately characterize the original

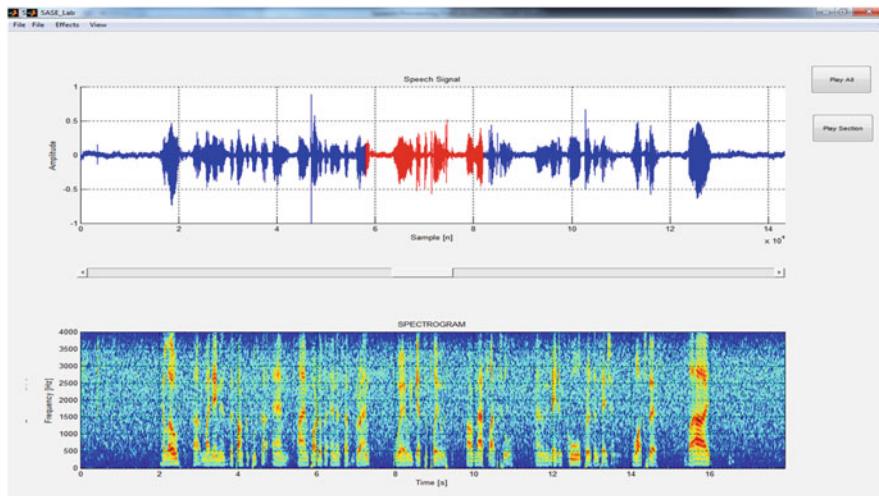


Fig. 24.21 3 seconds of data selected from the spectrogram depicted in red-color from the waveform

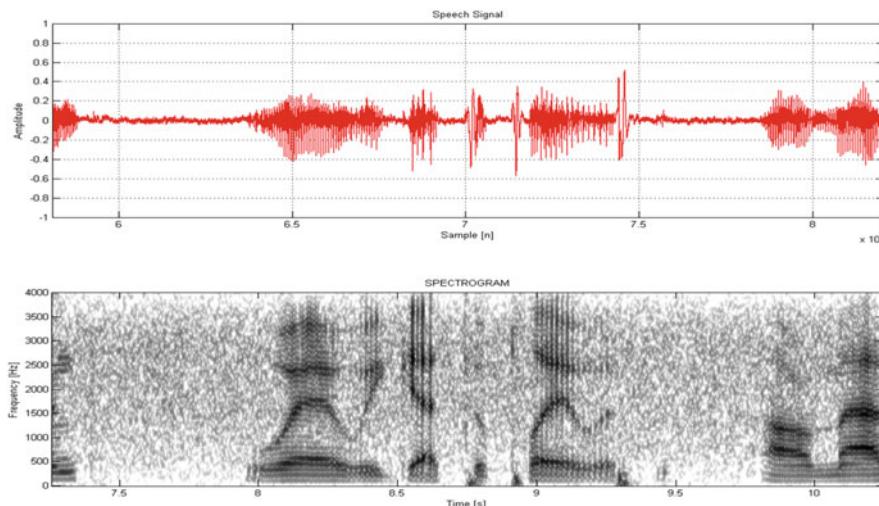


Fig. 24.22 3 seconds of data depicted with corresponding spectrogram computed from the selected section of waveform data

sound from the frame of speech. For different sounds of speech, the feature vector needs to show distinct characteristics to allow analysis and recognition of the original speech. The following five figures show how SASE Lab analyzes the vowel sounds “a, e, i, o, u”. These sounds are produced by voiced speech, in which, the vocal tract is driven by periodic pulses of air from the lungs. The periodic nature of the driving signal is characterized through the speaker’s pitch. For example,

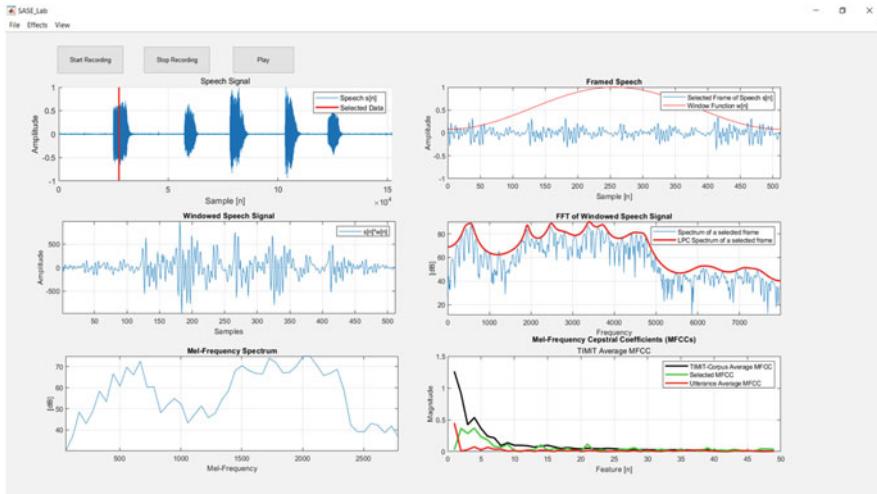


Fig. 24.23 Vowels, 'a', 'e', 'i', 'o', 'u' uttered and captured

males tend to have lower pitch than females and thus have a greater amount of time between each pulse of air. The pitch of the speaker can be observed on the frequency spectrum of the SASE Lab (Fig. 24.23).

In the next figure, the speech waveform was recorded for a speaker saying the sounds “a, e, i, o, u” (/ey/, /iy/, /ay/, /ow/, /uw/). This can be observed in the first plot that shows the entire speech signal. There are five distinct regions where there is significant amplitude data indicating sound. These five regions are surrounded by low amplitude signal-data, showing slight pauses between each sound uttered. In the first graph, a vertical red line is seen on the first region of sound, the “a” or /ey/ sound. The placement of the vertical red line controls which frame of the entire speech signal is to be analyzed. The next plot shows a periodic signal, with approximately three complete periods. This is the frame that has been selected for analysis. The next three plots show the signal as it passes through each stage of speech processing. The final plot on the bottom right, shows the mel-frequency cepstral coefficients (MFCCs) for the particular frame being analyzed. These mfcc values are called the feature vector where the plot depicts all 48 cepstral coefficients as computed by matlab. Note that, as displayed in the legend of that plot, graph contains three sub-plots, with black is the graph depicting TIMIT-Corpus average MFCC, with green the selected section of the speech is depicted, and finally with the red the utterance average is depicted. Note that the first value of feature vector is omitted from the plot due to the fact that matlab uses 1 index arrays. Also, this value contains the energy of the signal and typically is of far greater magnitude than the other values, hence, it is not shown on the plot as it would cause re-scaling of the graph and the detail of the other features to be lost. Finally, one additional curve is added to the FFT spectrum plot depicted in read in the 4 graph containing spectrum.

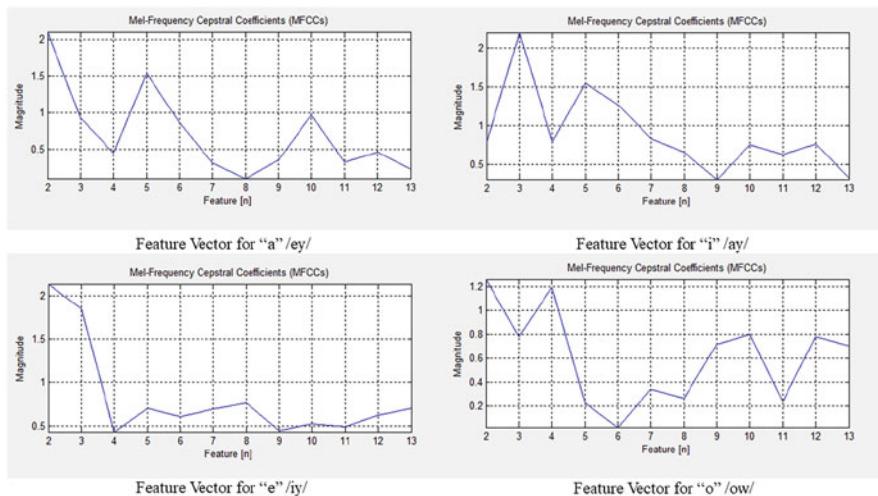


Fig. 24.24 Vowels, 'a', 'i', 'e', 'u'

This red curve depicts modeled spectrum by applying Liner-Prediction of order 20. As can be observed from the figure, the model performed as expected (Fig. 24.24).

The four figures above show how the feature vector differs for each sound produced. The difficulty lies in the fact that even for the same speaker, every time a particular sound is produced, there will be slight variations. This is one factor that makes speech recognition a complicated task. Current solutions require training a model for each sound. The training process entails collecting many feature vectors for a sound and creating a statistical model of the distribution of the features for the given sound. Then, when comparing an unknown feature vector to the likely distributions of features for a given sound, a probability that the unknown feature vector belongs to a known sound can be computed.

24.5 Wake-Up-Word Speech Recognition

Speech is considered one of the most natural forms of communications between people (Juang and Rabiner 2005). Spoken language has the unique property that it is naturally learned as part of human development. However, this learning process presents challenges when applied to digital computing systems.⁴

The goal of Automatic Speech Recognition (ASR) is to address the problem of building a system that maps an acoustic signal into a string of words. The idea of being able to perform speech recognition uttered from any speaker in any environment is still a problem that is far from being solved. However, recent advancements in the field have resulted in ASR systems that are applicable to some of Human Machine Interaction (HMI) tasks. ASR is already being successfully

applied in application domains such as telephony (automated caller menus) and monologue transcriptions for a single speaker. Several motivations for building ASR systems are, presented in order of difficulty, to improve human-computer interaction through spoken language interfaces, to solve difficult problems such as speech-to-speech translation, and to build intelligent systems that can process spoken language as proficiently as humans.

Speech as a computer interface has numerous benefits over traditional interfaces using mouse and keyboard: speech is natural for humans, requires no special training, improves multitasking by leaving the hands and eyes free, and is often faster and more efficient to transmit than the information provided using conventional input methods.

24.5.1 Introduction

In the presented work, the concept of Wake-Up-Word (WUW) is being introduced. The definition of the [Wake Up Word](#) task is presented in the next section. The implementation details and experimental evaluations of the WUW-SR system are depicted in the section depicting results. The WUW recognition paradigm partially applied to general word recognition is presented in the following section. Concluding remarks and future work is provided in the last section.

24.5.2 Wake-Up-Word Paradigm

In the recent developments (Kępuska and Klein [2009](#)) were focused on the Wake-up-Word (WUW) Speech Recognition paradigm. WUW has the following unique requirement: Detect a single word or phrase when spoken in an alerting context, while rejecting all other words, phrases, sounds, noises and other acoustic events with virtually 100% accuracy including the same word or phrase of interest spoken in a non-alerting (i.e. referential) context [Kępuska, V., 2011](#).

One of the goals of speech recognition is to allow natural communication between humans and computers via speech, where natural implies similarity to the ways humans interact with each other. A major obstacle to this is the fact that most systems today still rely to some extent on non-speech input, such as pushing buttons or a mouse clicks. However, much like a human assistant, a natural speech interface must be continuously listening and must be robust enough to recover from any communication errors without non-speech intervention. Problem with present SR system is that they solely rely on push-to-talk and non-speech intervention paradigms.

24.5.2.1 Push-to-Talk

Speech recognizers deployed in continuously listening mode are continuously monitoring acoustic input and do not necessarily require non-speech activation. This is in contrast to the push-to-talk model, in which speech recognition is only activated when the user “pushes a button”. Unfortunately, today’s continuously listening speech recognizers are not reliable enough due to their insufficient accuracy, especially in the area of correct rejection. For example, such systems often respond erratically, even when no speech is present. They sometimes interpret a background noise as speech, and they sometimes incorrectly assume that certain speech is addressed at the speech recognizer when in fact it is targeted elsewhere (context misunderstanding). These problems have traditionally been solved by the push-to-talk model: requesting the user to push a button immediately before or during talking or similar prompting paradigms. This action in fact represents an explicit triggering of the recognizer while in all other times the recognizer remains inactive, hence avoiding false triggers.

24.5.2.2 Non-Speech Intervention

Another problem with traditional speech recognizers is that they cannot recover from errors gracefully, and often require non-speech intervention. Any speech-enabled human-machine interface based on natural language relies on carefully crafted dialogues. When the dialogue fails, currently there is no good mechanism to resynchronize the communication, and typically the transaction between human and machine fails by termination. A typical example is a SR system which is in a dictation state, when in fact the human is attempting to use command-and-control to correct previous dictation error. Often the user is forced to intervene by pushing a button or keyboard key to resynchronize the system.

Current SR systems that do not deploy the push-to-talk paradigm use implicit context switching. For example, a system that has the ability to switch from “dictation mode” to “command mode” does so by trying to infer whether the user is uttering a command rather than dictating text. This task is rather difficult to perform with high accuracy, even for humans. The push-to-talk model uses explicit context switching, meaning that the action of pushing the button (or similar paradigm) explicitly sets the context of the speech recognizer to a specific state.

To achieve the goal of developing a natural speech interface, it is first useful to consider human-to-human communication. Upon hearing an utterance of speech a human listener must quickly make a decision whether or not the speech is directed to him or her. This decision determines whether the listener will make an effort to “process” and understand the utterance. Humans can make this decision quickly and robustly by utilizing visual, auditory, semantic, and/or additional contextual clues.

Visual clues might be gestures such as waving of hands or other facial expressions. Auditory clues are attention grabbing words or phrases such as the listener’s name (e.g. John), interjections such as “hey”, “excuse me,” and so forth. Addition-

ally, the listener may make use of prosodic information such as pitch and intonation, as well as identification of the speaker's voice.

Semantic and/or contextual clues are inferred from interpretation of the words in the sentence being spoken, visual input, prior experience, and customs dictated by the culture. Humans are very robust in determining when speech is targeted towards them, and should a computer SR system be able to make the same decisions its robustness would increase significantly.

24.5.2.3 Wake-Up-Word: Explicit Request

Wake-Up-Word is proposed as a method to explicitly request the attention of a computer using a spoken word or phrase (Kępuska, Elevator Simulator Screen Perspective, 2009a; Kępuska, Elevator Simulator User Perspective, 2009b). The WUW must be spoken in the context of requesting attention, i.e. alerting context and should not be recognized in any other context. After successful detection of WUW and its alerting context, the speech recognizer may safely interpret the following utterance as a command. The WUW is analogous to the button in push to talk, but the interaction is completely based on speech. Therefore it is proposed to use explicit context switching via WUW. Furthermore, this is similar to how context switching occurs in human to human communication as well.

24.5.3 Wake-Up-Word: Definition

Wake-up-word technology solves three major problems:

- Detecting Wake-Up-Word Context
- Identifying Wake-up-word
- Correct Rejection of Non-Wake-up-word
- Correct Rejection of Wake-up-word when uttered in referential context, that is, non-altering context.

24.5.3.1 Detecting Wake-Up-Word Context

The Wake-up-word system must be able to notify the host system that attention is required in certain circumstances and with high accuracy. Unlike keyword-spotting, see for example (Juang and Rabiner 2005), in which a certain keyword is recognized and reported during every occurrence, Wake-up-word dictates these occurrences only be reported during an alerting context. This context can be determined using features such as leading and trailing silence, difference in the long term average of speech features, and prosodic information (pitch, intonation, rhythm, etc.). This is still active research area (Kępuska and Chih-Ti 2010).

24.5.3.2 Identifying Wake-Up-Word

After identifying the correct context for a spoken utterance, the WUW paradigm shall be responsible for determining if the utterance contains the pre-defined Wake-up-Word to be used for command (e.g. “Computer”) with a high degree of accuracy, e.g., > 99% (K  puska and Klein [2009](#)).

24.5.3.3 Correct Rejection of Non-Wake-Up-Word

Similar to identification of the WUW, the system shall also be capable of filtering speech tokens that are not Wake-up-words with practically 100% accuracy to guarantee 0% false acceptances (K  puska and Klein [2009](#)).

24.5.4 Wake-Up-Word System

The concepts of Wake-Up-Word have been expanded in (K  puska and Klein [2009](#)). Currently, the system is implemented in C++, Objective C as well as Java language. It provides four major components for achieving the goals of Wake-Up-Word for use in a real-time environment.

24.5.4.1 Wake-Up-Word Front End

This system component is responsible for extracting features from the input audio signal. The current system is capable of extracting Mel-Filtered Cepstral Coefficients (MFCC), Linear Predictive Coding coefficients (LPC), and enhanced MFCC features.

24.5.4.2 Voice Activity Detector (VAD)

A large portion of the input audio signal to the system are non-speech events such as silence or environmental noise. Filtering this information is critical in order to ensure the system is only listening during speech events of interest. Areas of audio that are determined to be speech-related are then forwarded to the later stages of the Wake-Up-Word system.

24.5.4.3 Wake-Up-Word Back End

The Back End performs a complex recognition procedure based on Hidden Markov Models (HMMs). HMMs are continuous densities HMM's.

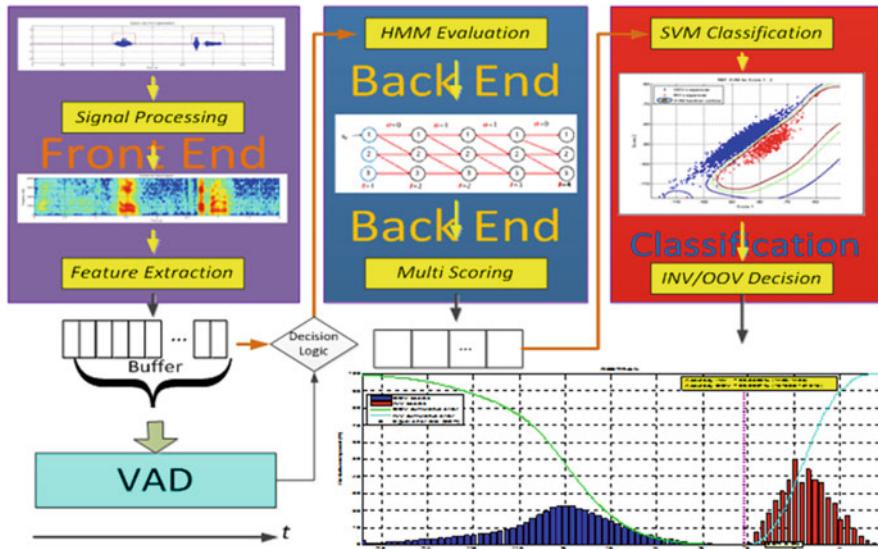


Fig. 24.25 Wake-up-word speech recognition system

24.5.4.4 Support Vector Machines Classification

The final system component is responsible for classifying speech signals as In-Vocabulary (INV) or Out-of-Vocabulary (OOV) using Support Vector Machines (SVMs). In the Wake-up-word context, the only INV word is the one selected for command and control of the host system. Any other word or sound is classified as OOV.

The following diagram illustrates the top-level workflow of the WUW system (Fig. 24.25):

24.5.5 *Front-End of the Wake-Up-Word System*

The front-end is responsible for extracting features from the input signal. Three sets of features are extracted: Mel-Filtered Cepstral Coefficients (MFCC) feature, LPC (Linear Predictive Coding) feature, smoothed MFCCs, and non-linearly Enhanced MFCCs.

The following image depicts a waveform superimposed with its VAD segmentation, its spectrogram, and its enhanced spectrogram (Fig. 24.26).

The details of computing MFCC features are covered in the section *refSpeech Processing*. The novelty is that we deploy the LPC (Linear Predictive Coding), smoothed MFCCs, and Enhanced MFCCs, as described in (K  puska and Klein 2009). Those triple features are used by the Back End to score each frame each

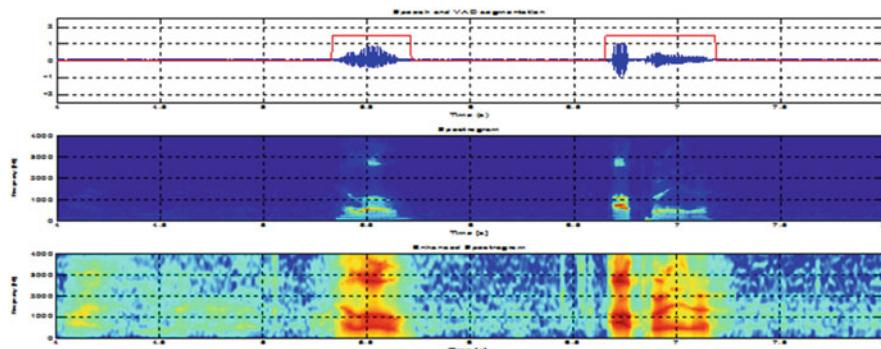


Fig. 24.26 Speech signal with voice activity detector segmentation, spectrogram, and enhanced spectrogram generated by the front end module (K  puska and Klein 2009)

with its corresponding HMM model. The special care was taken in designing VAD as it is crucial for it to perform perfectly.

24.5.5.1 VAD Classification

In the first phase, for every input frame VAD decides whether the frame is speech-like or non-speech-like. Several methods have been implemented and tested to solve this problem.

In the first implementation, the decision was made based on three features: log energy difference, spectral difference, and MFCC difference. A threshold was determined empirically for each feature, and the frame was considered speech-like if at least two out of the three features were above the threshold. This was in effect a Decision Tree classifier, and the decision regions consisted of hypercubes in the feature space.

In order to improve the VAD classification accuracy, research has been carried out to determine the ideal features to be used for classification. Hence, Artificial Neural Networks (ANN) and Support Vector Machines (SVM) were tested for automatic classification. One attempt was to take several important features from a stream of consecutive frames and classify them using ANN or SVM. The idea was that the classifier would make a better decision if shown multiple consecutive frames rather than a single frame. The result, although good, was too computationally expensive, and the final implementation still uses information from only a single frame.

24.5.5.2 First VAD Phase—Single Frame Decision

The final implementation uses the same three features: log energy difference, spectral difference, and MFCC difference; however, classification is performed using a linear SVM. There are several advantages over the original method. First, the classification boundary in the feature space is a hyperplane, which is more robust than the hypercubes produced by the decision tree method. Second, the thresholds do not have to be picked manually but can be trained automatically (and optimally) using labeled input files. Third, the sensitivity can be adjusted in smooth increments using a single parameter: the SVM decision threshold. Recall that the output of a SVM is a single scalar, $u = wx - b$ (Klein 2007). Usually the decision threshold is set at $u = 0$, but it can be adjusted in either direction depending on the requirements. Finally, the linear SVM kernel is extremely efficient, because classification of new data requires just a single dot-product computation (Fig. 24.27).

In the presented figure, the red points correspond to frames contain speech while the blue points correspond to non-speech frames, as labeled by a human listener. It can be seen that the linear classifier produces a fairly good separating plane between the two classes, and the plane could be moved in either direction by adjusting the threshold.

24.5.5.3 Second VAD Phase—Final Decision Logic

In the second phase, the VAD keeps track of the number of frames marked as speech and non-speech and makes a final decision. There are four param-

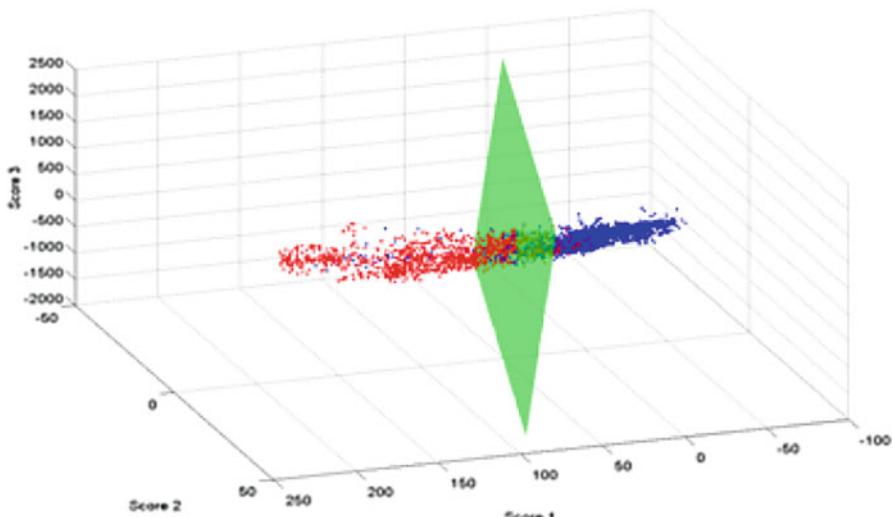


Fig. 24.27 Support vector machine decision surface as computed with linear SVM

eters: *MIN_VAD_ON_COUNT*, *MIN_VAD_OFF_COUNT*, *LEAD_COUNT*, and *TRAIL_COUNT*. The algorithm calls for a number of consecutive frames to be marked as speech in order to set the state to *VAD_ON*; this number is specified by *MIN_VAD_ON_COUNT*. It also requires a number of consecutive frames to be marked as non-speech in order to set the state to *VAD_OFF*; this number is specified by *MIN_VAD_OFF_COUNT*. Because the classifier can make mistakes at the beginning and the end, the logic also includes a lead-in and a trail-out time. When the minimum number of consecutive speech frames has been observed, VAD does not indicate *VAD_ON* for the first of those frames. Rather it selects the frame that was observed a number of time instances earlier; this number is specified by *LEAD_COUNT*. Similarly, when the minimum number of non-speech frames has been observed, VAD waits an additional number of frames before changing to *VAD_OFF*, specified by *TRAIL_COUNT*.

24.5.5.4 Back-End—Plain HMM Scores

The Back End is responsible for scoring observation sequences. The WUW-SR system uses a Hidden Markov Models for acoustic modeling, and as a result the back end consists of a HMM recognizer. Prior to recognition, HMM model(s) must be created and trained for the word or phrase which is selected to be the Wake-Up-Word.

When the VAD state changes from *VAD_OFF* to *VAD_ON*, the HMM recognizer resets and prepares for a new observation sequence. As long as the VAD state remains *VAD_ON*, feature vectors are continuously passed to the HMM recognizer, where they are scored using the novel triple scoring method. If using multiple feature streams, recognition is performed for each stream in parallel. When VAD state changes from *VAD_ON* to *VAD_OFF*, multiple scores (e.g., MFCC, LPC and E-MFCC Score) are obtained from the HMM recognizer and are sent to the SVM classifier. SVM produces a classification score which is compared against a threshold to make the final classification decision of *INV* or *OOV*.

For the first tests on speech data, a HMM was trained on the word “operator.” The training sequences were taken from the CCW17 and WUW-II (Kępuska and Klein 2009) corpora for a total of 573 sequences from over 200 different speakers. After features were extracted, some of the erroneous VAD segments were manually removed. The INV testing sequences were the same as the training sequences, while the OOV testing sequences included the rest of the CCW17 corpus (3833 utterances, 9 different words, over 200 different speakers). The HMM was a left-to-right model with no skips, 30 states, and 6 mixtures per state, and was trained with two iterations of Baum-Welch.

The score is the result of the Viterbi algorithm over the input sequence. Recall that the Viterbi algorithm finds the state sequence that has the highest probability of being taken while generating the observation sequence. The final score is that probability normalized by the number of input observations, T . Figure 24.8 below shows the result.

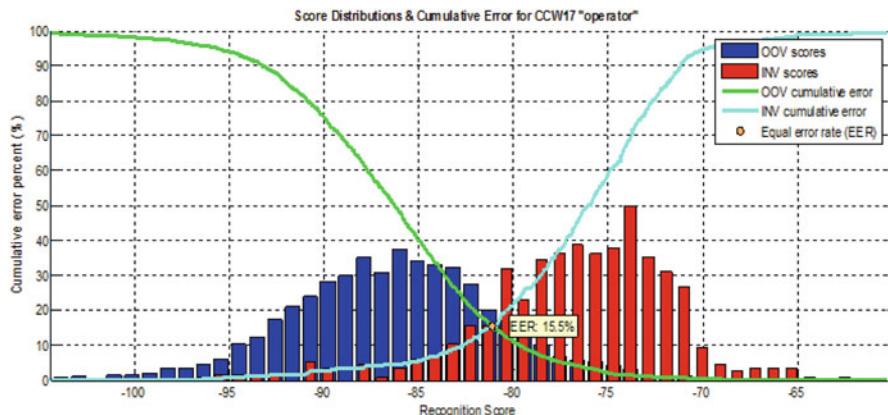


Fig. 24.28 Scoring of plain WUW recognition

The distributions look Gaussian, but there is significant overlap between them. The equal error rate of 15.5% essentially means that at that threshold, 15.5% of the OOV words would be classified as INV, and 15.5% of the INV words would be classified as OOV. Obviously, no practical applications can be developed based on the performance of this recognizer (Fig. 24.28).

Clearly, classic approach to Speech Recognition is not going to be able to lead to goal of having close to 0% error rate, hence an innovative way of scoring was invented and presented next.

24.5.5.5 SVM Classification

After HMM recognition, the algorithm uses two additional scores for any given observation sequence (e.g., MFCC, LPC and e-MFCC). When considering the three scores as features in a three dimensional space, the separation between INV and OOV distributions increases significantly. The next experiment runs recognition on the same data as above, but this time the recognizer uses the triple scoring algorithm to output three scores: Score 1, Score 2, and Score 3 (Kępuska and Klein 2009).

Triple-Scoring Method The figures below show two-dimensional scatter plots of Score 1 vs. Score 2, and Score 1 vs. Score 3 for each observation sequence (e.g., MFCC, LPC and e-MFCC). In addition, a histogram on the horizontal axis shows the distributions of Score 1 independently, and a similar histogram on the vertical axis shows the distributions of Score 2 and Score 3 independently. The histograms are hollowed out so that the overlap between distributions can be seen clearly. The distribution for Score 1 is exactly the same as in the previous section, as the data and model haven't changed. Any individual score does not produce a good separation between classes, and in fact the Score 2 distributions have almost complete overlap. However, the two dimensional separation in either case is remarkable. When all

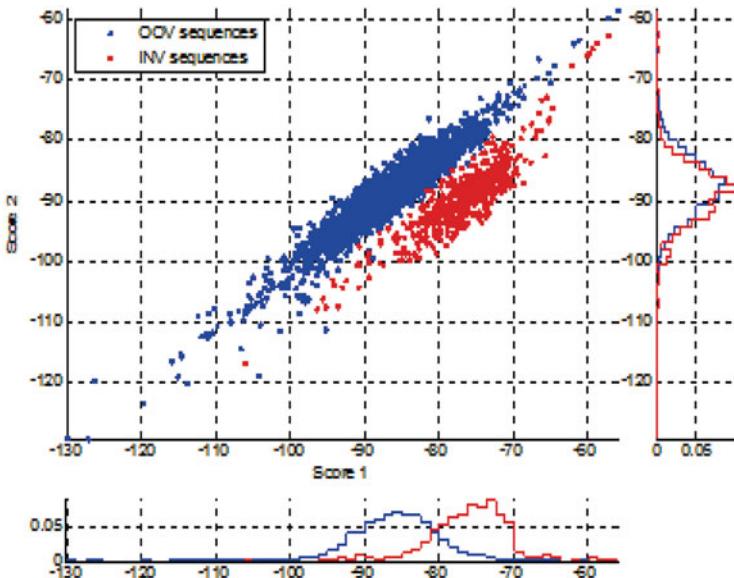


Fig. 24.29 Score1 vs. Score2 example (Kępuska and Klein 2009)

three scores are considered in a three dimensional space, their separation is even better than either two dimensions as depicted in Figs. 24.29 and 24.30.

In order to automatically classify an input sequence as INV or OOV, the triple score feature space, \mathfrak{R}^3 , can be partitioned by a binary classifier into two regions: \mathfrak{R}_{+1}^3 and \mathfrak{R}_{-1}^3 . The SVMs have been selected for this task because of the following reasons: they can produce various kinds of decision surfaces including radial basis function, polynomial, and linear; they employ Structural Risk Minimization (SRM) (Burges 1998) to maximize the margin which has shown empirically to have good generalization performance.

SVM Parameters Two types of SVMs have been considered for this task: linear and RBF. The linear SVM uses a dot product kernel function, $K(x, y) = x \cdot y$, and separates the feature space with a hyperplane. It is very computationally efficient because no matter how many support vectors are found, evaluation requires only a single dot product. Presented Fig. 24.31 shows that the separation between distributions based on Score 1 and Score 2 is almost linear, so a linear SVM would likely give good results. However, in the Score 1/Score 3 space (Fig. 24.32), the distributions have a curvature, so the linear SVM is unlikely to generalize well for unseen data. The figures below show the decision boundary found by a linear SVM trained on Score 1+2, and Score 1+3, respectively.

The line in the center represents the contour of the SVM function at $u = 0$, and outer two lines are drawn at $u = \pm 1$. Using 0 as the threshold, the accuracy of Scores 1–2 is 99.7% Correct Rejection (CR) and 98.6% Correct Acceptance (CA),

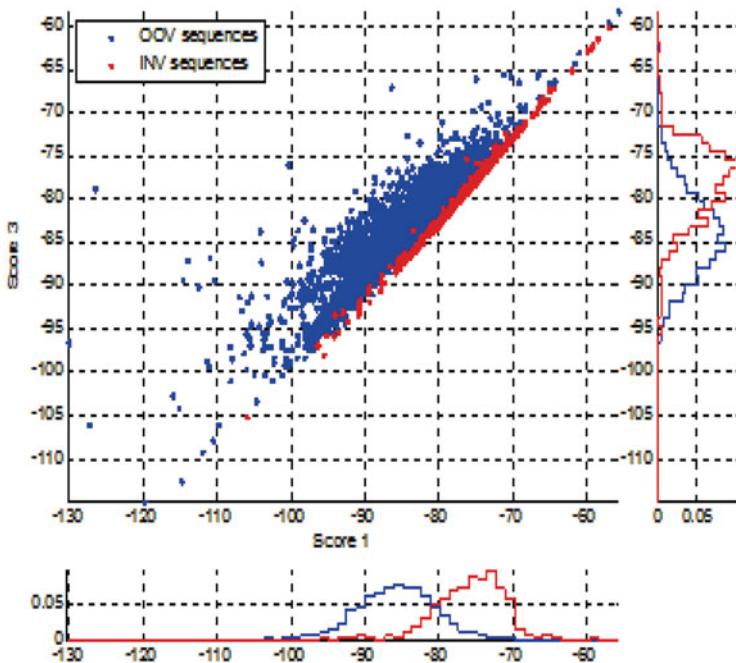


Fig. 24.30 Score1 vs. Score3 example (K  puska and Klein 2009)

while for Scores 1–3 it is 99.5% CR and 95.5% CA. If considering only two features, Scores 1 and 2 seem to have better classification ability. However, combining the three scores produces the plane shown below from two different angles (Fig. 24.33).

The plane split the feature space with an accuracy of 99.9% CR and 99.5% CA (just 6 of 4499 total sequences were misclassified). The accuracy was better than any of the 2 dimensional cases, indicating that Score 3 contains additional information not found in Score 2. The classification error rate of the linear SVM is shown below (Figs. 24.34, and 24.35):

The conclusion from the presented experiment is that using the triple scoring method combined with a linear SVM decreased the equal error rate on this particular data set from 15.5% to 0.2%, or in other words increased accuracy by over 76.5 times (i.e., error rate reduction of 7650%)!

Radial Basis Function & SVM In the next experiment, a Radial Basis Function (RBF) kernel was used for the SVM. The RBF function, $K(x, y) = e^{-\gamma|x-y|^2}$, maps feature vectors into an infinitely dimensional Hilbert space and is able to achieve complete separation between classes in most cases. However, the γ parameter must be chosen carefully in order to avoid overtraining. As we have found no way off determining it automatically, a grid search may be used to find a good value. For most experiments $\gamma = 0.008$ gave good results. Shown below are the RBF SVM contours for both two-dimensional cases.

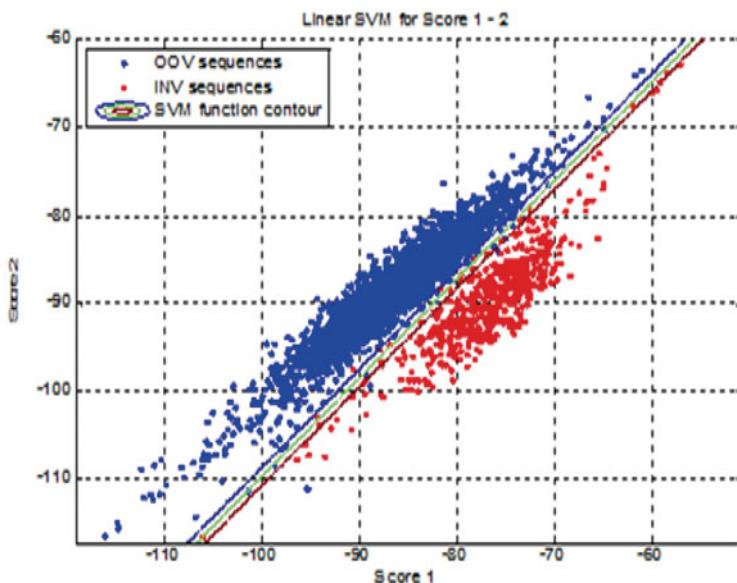


Fig. 24.31 Score1 vs. Score2 linear discrimination surface (K  puska and Klein 2009)

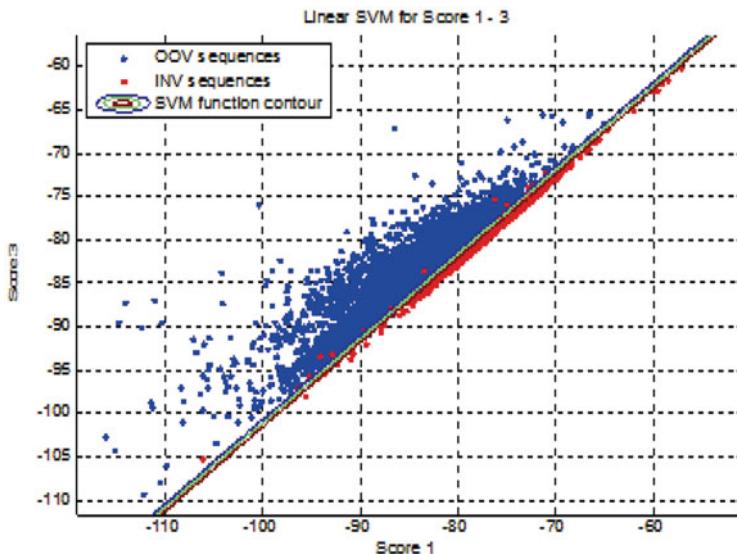


Fig. 24.32 Score1 vs. Score3 linear discrimination surface (K  puska and Klein 2009)

At the $u = 0$ threshold, the classification accuracy was 99.8% CR, 98.6% CA for Score 1-2, and 99.4% CR, 95.6% CA for Score 1-3. In both cases the RBF

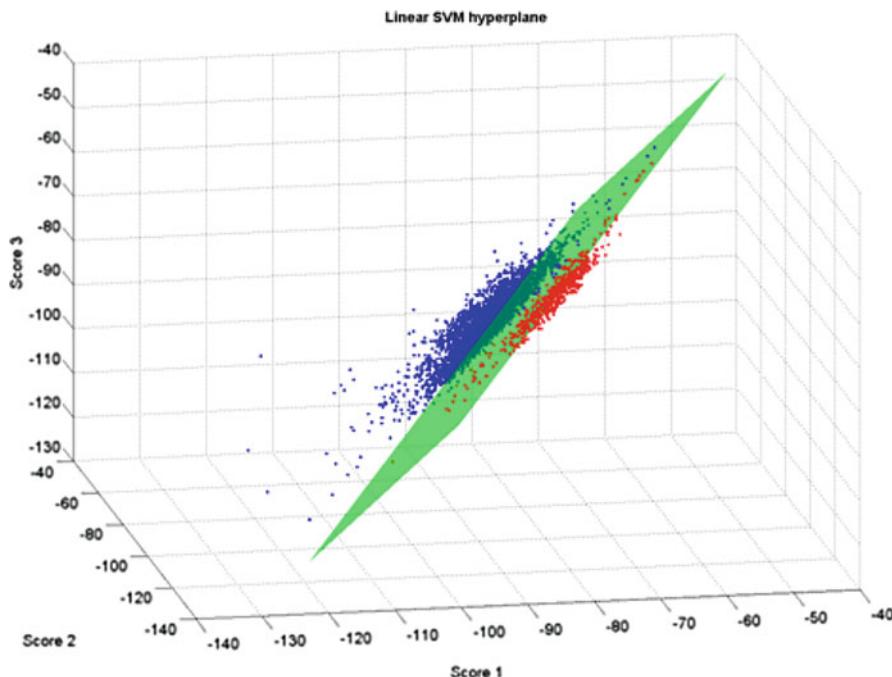


Fig. 24.33 Score1 vs. Score2 vs. Score3 hyper-plane discrimination surface (K  puska and Klein 2009)

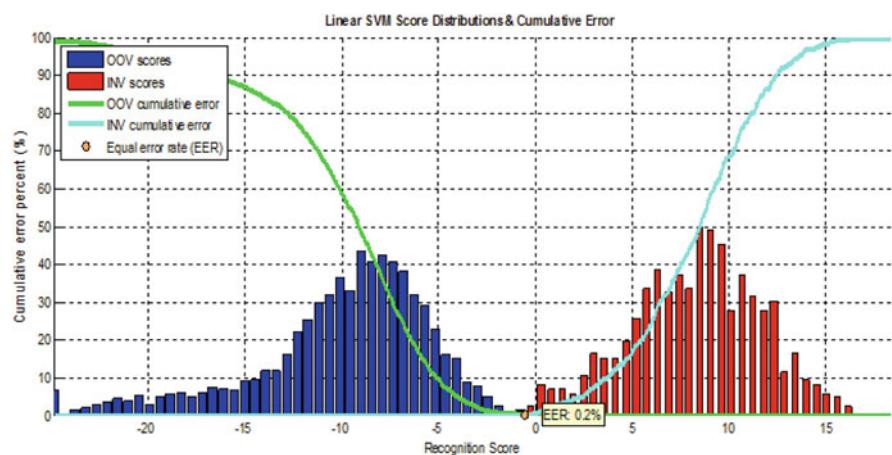


Fig. 24.34 Distribution of Scores for triple-scoring and linear SVM (K  puska and Klein 2009)

kernel formed a closed decision region around the INV points (Recall that the SVM decision function at $u = 0$ is shown by the green line).

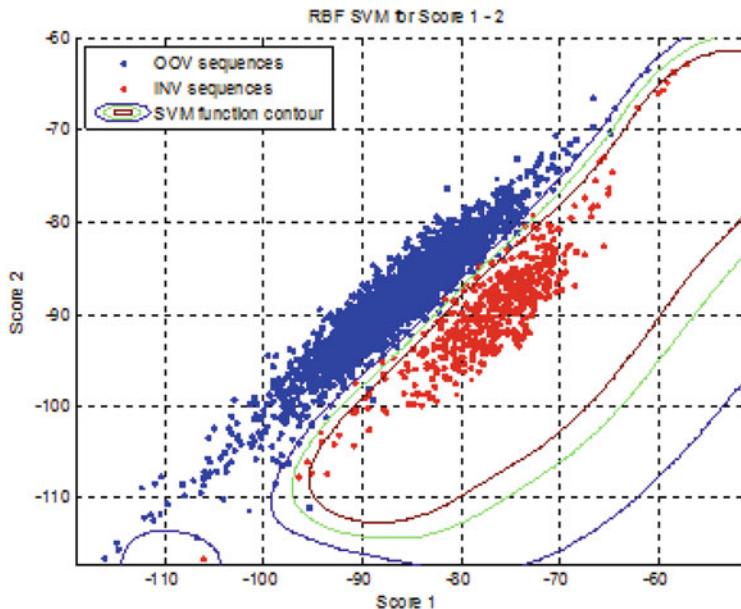


Fig. 24.35 Scatter plot of INV and OOV distributions of Scores1 vs Score2 with RBF discriminating function (Kępuska and Klein 2009)

Some interesting observations can be made from these plots. First, the INV outlier in the bottom left corner of the first plot caused a region to form around it. SVM function output values inside the region were somewhere between -1 and 0 , not high enough to cross into the INV class. However, it is apparent that the RBF kernel is sensitive to outliers, so the γ parameter must be chosen carefully to prevent overtraining. Had the γ parameter been a little bit higher, the SVM function output inside the circular region would have increased beyond 0 , and that region would have been considered INV (Fig. 24.36).

Second, the RBF kernel's classification accuracy showed almost no improvement over the linear SVM. However, it is expected that due to the RBF kernel's ability to create arbitrary curved decision surfaces, it will have better generalization performance than the linear SVM's hyperplane.

The figure below shows a RBF kernel SVM trained on all three scores (Fig. 24.37):

The RBF kernel created a closed 3-dimensional surface around the INV points and had a classification accuracy of 99.95% CR, 99.30% CA. If considering $u = 0$ as the threshold, the triple score SVM with RBF kernel function shows only little improvement over the linear SVM for this data set. However, as shown below, the SVM score distributions are significantly more separated, and the equal error rate is lower than the linear SVM; from 0.2% to 0.1% (Fig. 24.38).

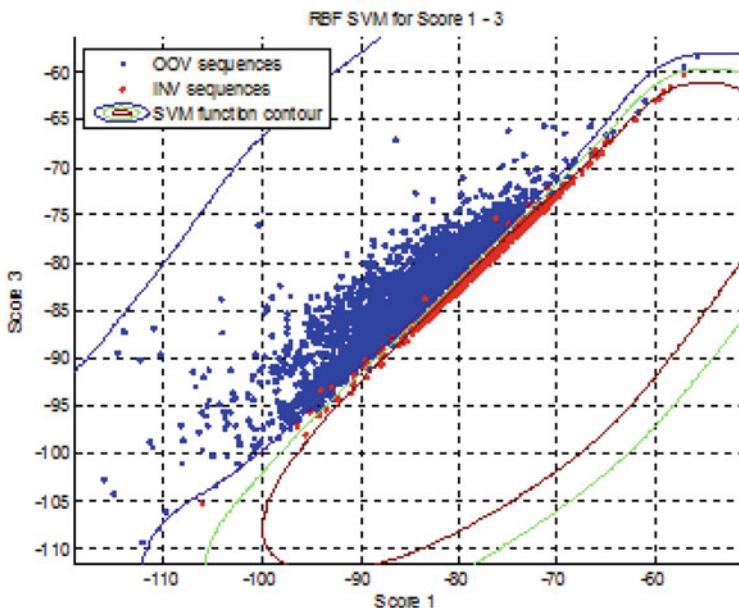


Fig. 24.36 Scatter plot of INV and OOV distributions of Scores1 vs Score3 with RBF discriminating function (Kępuska and Klein 2009)

There Is “no Data like More Data”

Final results when combining all three features using all the available data for testing and training (Callhome, Phonebook, WUW and WUWII Corpora, CCW17) provides a clear superiority of the presented method (presented below). In this test the INV accuracy of only two (2) errors out of 1425 ‘operator’ utterances or 99.8596% and OOV accuracy of twelve (12) errors on 151615 tokens or 99.9921%.

Overall recognition rate of WUW-SR utilizing Callhome, Phonebook, WUW, WUWII and CCW17 corpora was presented (Fig. 24.39). From the chart it clearly demonstrates that our WUW-SR’s improvement over current state of the art recognizers. WUW-SR with three feature streams was several orders of magnitude superior to the baseline recognizers in INV recognition and particularly in OOV rejection. Of the two OOV corpora, Phonebook was significantly more difficult to recognize for all three systems due to the fact that every utterance was a single isolated word. Isolated words have similar durations to the WUW and differ just in pronunciation, putting to the test the raw recognition power of ASR system. HTK had 8793 false acceptance errors on Phonebook and the commercial SR system of 2009 had 5801 errors.

WUW-SR demonstrated its OOV rejection capabilities by committing just total of 12 false acceptance errors on all the corpora (151615 tokens) used for OOV rejection testing while maintaining high recognition rate by committing only 2 false rejection errors for 1425 INV words. This result is not being biased to optimize

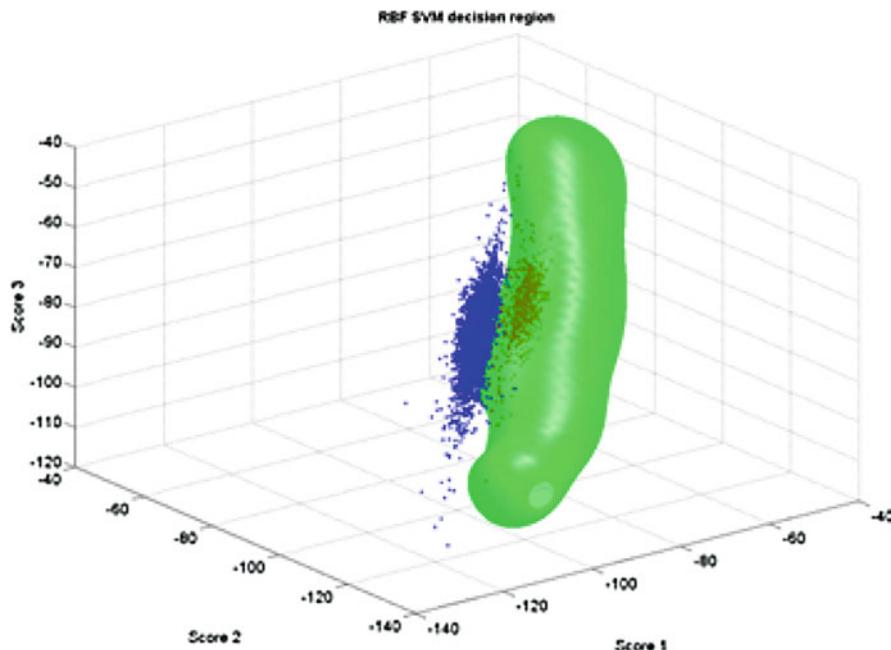


Fig. 24.37 Discriminating surface presented in 3D (Kępuska and Klein 2009)

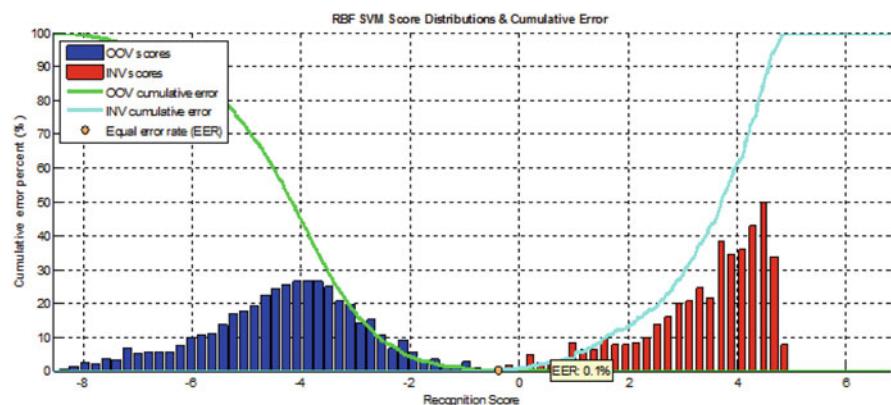


Fig. 24.38 PDF of INV and OOV distributions for RBF (Kępuska and Klein 2009)

against neither false rejection nor false acceptance. If biasing is necessary it can be easily accomplished by shifting the threshold factor of SVM from zero toward -1 to reduce false rejection or toward $+1$ to reduce false acceptance.

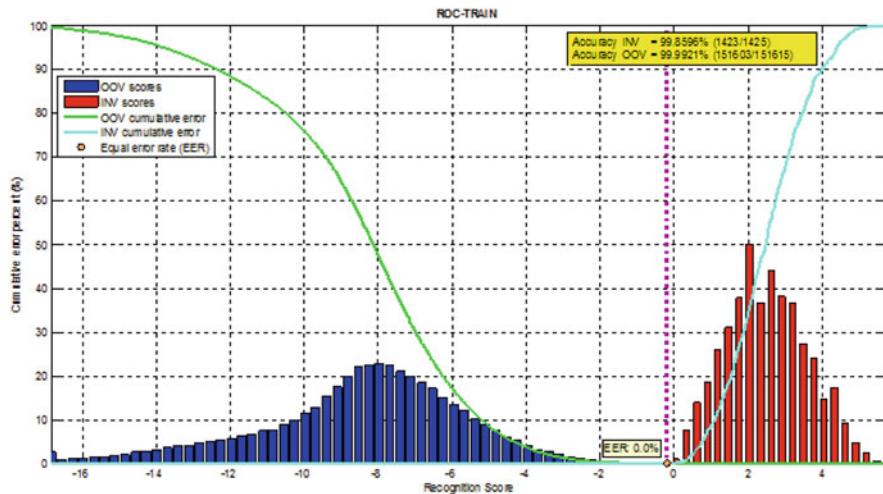


Fig. 24.39 PDF of INV and OOV distributions for RBF utilizing more data (K  puska and Klein 2009)

24.6 Conclusion

The WUW-SR system developed in this work provides for efficient and highly accurate speaker independent recognitions at performance levels not achievable by current state of the art recognizers. Extensive testing demonstrates accuracy improvements superior by several orders of magnitude over the best known academic speech recognition system, HTK, as well as a leading commercial speech recognition system. Specifically, the WUW-SR system correctly detects the WUW with 99.98% accuracy. It correctly rejects non-WUW with over 99.99% accuracy. The WUW system makes 12 errors in 151615 words or less than 0.008%. Assuming speaking rate of 100 words per minute it would make 0.47 false acceptance errors per hour, or one false acceptance in 2.1 hours.

At the time when this document was written, at around the end first decade of 21'st century, comparison of WUW performance in detection and recognition performance was 2525%, or 26 times better than HTK for the same training and testing data, and 2,450%, or 25 times better than Microsoft SAPI 5.1 recognizer. The out-of-vocabulary rejection performance is over 65,233%, or 653 times better than HTK, and 5900% to 42,900%, or 60 to 430 times better than the Microsoft SAPI 5.1 recognizer (Fig. 24.40).

In order to achieve these levels of accuracy, the following innovations were accomplished:

- Hidden Markov Model triple scoring with Support Vector Machine classification
- Combining multiple speech feature streams (MFCC, LPC-smoothed MFCC, and Enhanced MFCC)
- Improved Voice Activity Detector with Support Vector Machines classification

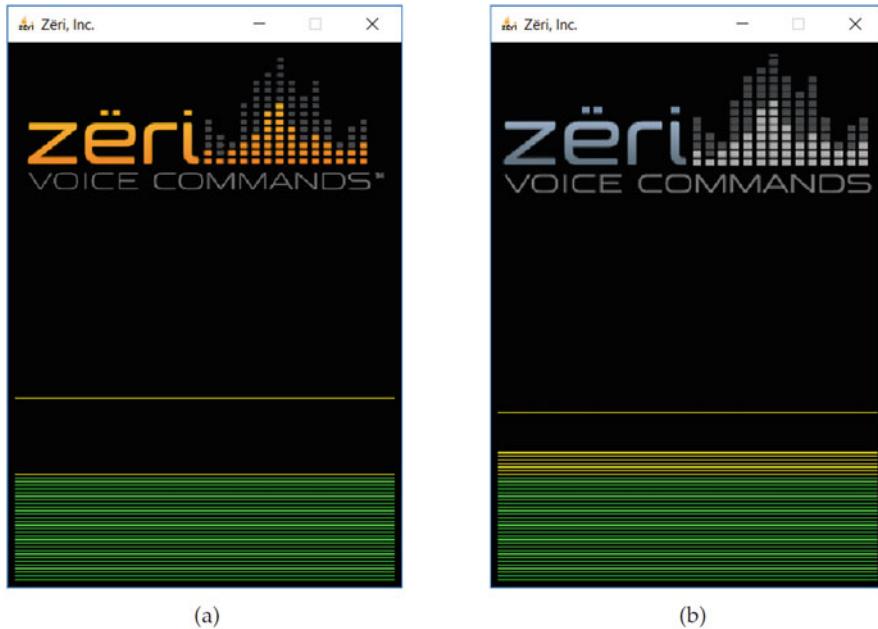


Fig. 24.40 Example of “no Wake-up-Word”/some other word spoken, and the case when “Wake-up-Word” was spoken not triggering off and triggering the system as indicated with different color as well as audibly played sound “yes”. **(a)** Wake-up-word demo “off”. **(b)** Wake-up-word demo “on”

24.6.1 Wake-Up-Word: Tool Demo

The following section is depicting our WUW package written in C#.

Due to the fact that present publishing technology is not capable of incorporating and displaying a movie-clip captured from a computer, we are providing the link where the snapshot of the demo can be viewed “[Operator](#)” by [clicking the word](#).

In addition, we are providing pictures that captures static screen depicting a VU meter, as in Fig. 24.40a) provided below depicting operation of the VU meeter without being triggered by WUW, and upon hearing the selected WUW word (e.g., Operator) spoken in alerting context (and not in referential context) is depicted in Fig. 24.40b displaying response of the system with synthesised word “yes”. This example utilizes the to VU meter tool provided as freeware written in java programming language. In addition, the wrapper uses C# code that was specifically written for WUW. The WUW code was originally written in C++ programming language.

24.6.2 *Elevator Simulator*

The same technology utilizing WUW is demoed in voice activated elevator simulator; shown in you-tube via the links provided below:

Wake Up Word Elevator Demo—User View—YouTube

<https://www.youtube.com/watch?v=j5CeVtQMvK0>

Wake Up Word Elevator Demo—Screen View—YouTube

https://www.youtube.com/watch?v=OQ8eyBTbS_E

The snapshot of the tool are displayed in the following figures (Fig. 24.41):

24.7 Background Information

This chapter is dedicated to bringing all different aspect of machine learning to a few practical examples that Dr. Veton Këpuska, his groups, teams and students have developed through out the years. The aim is to improve understanding of our audience in the area of how humans communicate via spoken language.

General

Speech processing entails many different aspects of mathematics and signal processing techniques. The main goal of this process is a reduction in amount of data while maintaining an accurate representation of the speech signal characteristics. For every frame of speech, typically 10–30 milliseconds, a feature vector must be computed that contains these characteristics. An average frame size is approximately 256 samples of audio data, while the feature vector typically is only 13 values. This reduction in amount of data allows for more efficient processing of the feature vector. For example, if the feature vector is being passed along to a speech recognition process, an analysis on the feature vector will be computationally more efficient than analysis on the original frame of speech.

The outline of a speech processing system contains several stages. The first stage is to separate the speech signal into frames of 10–30 millisecond duration. Speech is typically constant over this period, and allows for efficient analysis of a semi-stationary signal. This frame of data is then passed through the other stages of the process to produce the end result, a feature vector. The next step is to apply a pre-emphasis filter to compensate for lower energy in the higher frequencies of human speech production. After this filter, the Fourier Transform of the signal is taken to compute the frequency spectrum of the speech frame. This information indicates what frequency content composed the speech signal. The frequency spectrum is then converted to a logarithmic scale through the process of mel-filtering. This step models the sound as humans perceive frequencies, logarithmically. After this step, the base 10 logarithm is taken of the resulting values from the previous step and finally, the discrete cosine transform is applied. The resulting vector is truncated to 13 values and forms the feature vector. This feature vector characterizes the type of

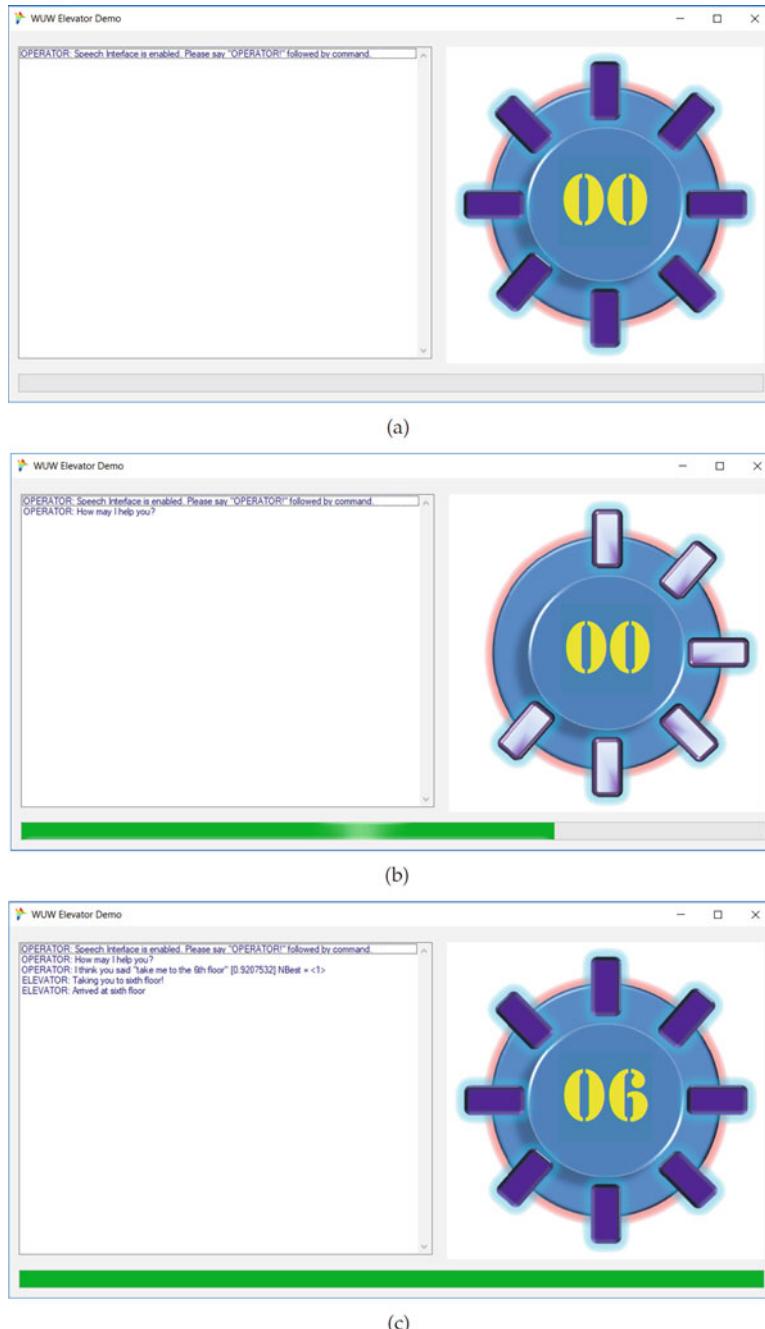


Fig. 24.41 Example of usage of WUW technology in elevator simulator. (a) Initial state of the elevator simulator. (b) Elevator simulator acquiring voice command. (c) Elevator simulator arriving at the destination floor

speech sounds present in the original frame, but with the advantage of using far less data. The feature vector can then be passed along to a speech recognition system for further analysis.

Suggested

The MATLAB tool created for this project, SASE Lab, performs all stages of speech processing to produce a feature vector for each frame of speech signal data. SASE Lab also shows graphs of data after each stage of the speech processing task. This breakdown of information allows the user to visualize how the data is manipulated through each step of the process. In addition to speech processing, the tool also incorporates several digital signal processing techniques to add audio effects to a speech signal. The application of these effects can then be analyzed for how they affect the feature vectors produced or at any stage of speech processing. Effects include echo, reverberation, flange, chorus, vibrato, tremolo, and modulation. Recommended Download : “SASE_Lab” from the provided link (this is required for the exercises: (https://fletch-my.sharepoint.com/:u/g/personal/vkepuska_fit_edu/EaphGzjeOlFpbIE3MgMlboBSOIjHDFwqvzfXF1q8eSCmg?e=MHvUSR).

For more information, readers are encouraged to read references used in this chapter Oppenheim et al. (2010), Phillips and John M. Parr (2008), and Quatieri (2002).

24.8 Exercises

- (1) Download “SASE_Lab” from the provided link: (https://fletch-my.sharepoint.com/:u/g/personal/vkepuska_fit_edu/EaphGzjeOlFpbIE3MgMlboBSOIjHDFwqvzfXF1q8eSCmg?e=MHvUSR)
- (2) Use TIMIT utterances sa1, provided in root directory of the SASE_Lab as indicated in this picture (Fig. 24.42):
- (3) Utilize the tool as described in section”

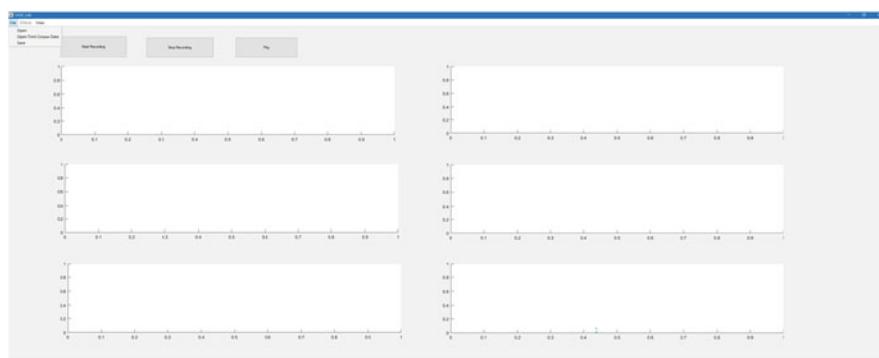


Fig. 24.42 Executing SASE_LAB from MATLAB

24.9 Speech Analysis and Sound Effects Laboratory (SASE_Lab)"

- Kępuska, Dr. Veton. Lecture notes can be obtained using the following link: https://fltech-my.sharepoint.com/personal/vkepuska_fit_edu/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fvkepuska%5Ffit%5Fedu%2FDocuments%2FClasses%2FECE5526 that contains the following:
 - Discrete Time Signal Processing Framework.
Ch2-Discrete-Time Signal Processing Framework2.pptx
 - Acoustics of Speech Production
Ch4-Acoustics_of_Speech_Production.pptx
 - Speech Signal Representations
Ch3-Speech_Signal_Representations.pptx
 - Automatic Speech Recognition
Ch5-Automatic Speech Recognition.ppx

References

- [Oppenheim2010] Oppenheim, Alan V., and Ronald W. Schafer. Discrete-time signal processing. 3rd ed. Upper Saddle River: Pearson, 2010.
- [Phillips2008] Phillips, Charles L., and John M. Parr. Signals, systems, and transforms. 4th ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.
- [Quatieri2002] Quatieri, T. F.. Discrete-time speech signal processing: principles and practice. Upper Saddle River, NJ: Prentice Hall, 2002.
- [Juang2005] Juang and Rabiner. Automatic Speech Recognition - A Brief History of the Technology Development, https://web.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf January 2005
- [Kepuska2009] Kępuska, V.Z. and Klein, T.B. (2009) A novel wake-up-word speech recognition system, wake-up-word recognition task, technology and evaluation, Nonlinear Analysis: Theory, Methods Applications, 71, e2772–e2789. <https://doi.org/10.1016/j.na.2009.06.089>
- [Kepuska2009a] Kępuska, V., Elevator simulator screen perspective, 2009. The link of this is in http://www.youtube.com/watch?v=OQ8eyBTbS_E. Accessed 11 September 2010
- [Kepuska2009b] Kępuska, V., Elevator simulator user perspecive, 2009. The link is <http://www.youtube.com/watch?v=j5CeVtQMyK0>. Accessed 11 September 2010.
- [Kepuska2010] Kępuska, V., D. S. Carstens, D. S. and Wallace, R. Leading and trailing silence in wake-up-word speech recognition, Proc. Int'l Conf. Industry, Eng, October, 2010
- [Kepuska2011] Kępuska, V., Wake-Up-Word Speech Recognition, Speech Technologies, IntechEditors: Ivo Ipsic June, 2011, <https://doi.org/10.5772/16242>
- [Burges1998] Burges, C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2, 121–167, (1998).

Chapter 25

Biomedical Signals: ECG, EEG



Overview

To understand the applications in the area of bio-medicine, some background technical knowledge is required. Possible applications need familiarity with the area. We address the chapter with such persons in mind. Originally, the topic is a transthoracic interpretation of the electrical activity of the heart over a period of time, as detected by electrodes. The heart sends out signals in the form of electrical impulses. Heart sounds are good examples of periodic signals.

In the speech section we introduced psychoacoustic phenomena. They contain many types of information and most of them are conveyed by the sender for certain purposes. In this and the next chapter, the theory previously covered play significant role. The difference is that the sender has no specific communication intention towards other humans. Nevertheless these phenomena contain a certain and sometimes important information about the sender. In this chapter the sender is the human body while in the next chapter it will be the earth. An example in both cases is silence. If no signal is sent for a certain period of time, in medicine the implications can be of maximal importance.

In this chapter we consider signals recorded from measurements of the human body. Such signals contain hidden information about the health of the persons. The single sample is not really informative; one rather needs a whole signal process. This process is stochastic because the human body is not a simple enough machine where one can predict the values. If one does two consecutive measurements while keeping all currently controllable variables identical, the results will not be exactly the same.

The living organisms consists of many component systems and each system is made up of several subsystems that carry on many physiological processes. Most physiological processes are accompanied by or manifest themselves by signals that reflect their nature and activities. The problem from the viewpoint of medicine is to extract the relevant information from the measurements. The techniques for this are in principle close to those employed for psychoacoustic phenomena in speech.

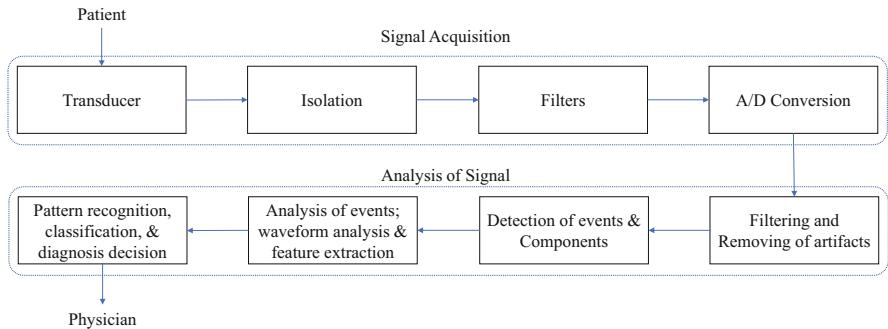


Fig. 25.1 Depiction of the typical process involving a patient, a machine, a physician

This is again connected with feature extraction. We consider two types of such measurements and signals, the ECG (i.e., Electrocardiogram) signals and the EEG (i.e., Electroencephalogram) signals. One connects with the heart and the other one with the brain. In medicine, one has to detect a pathology and design a therapy based on these measurements. Figure 25.1 visualizes the whole system.

In Fig. 25.1, one sees how medical activities rely on signals. It is important to realize that not only the interpretation of signals but often also their acquisition provides a major difficulty. The figure is quite simple. Each box contains a number of more or less sub boxes. In addition, one has the visual impressions of the medical expert that also influence the diagnosis. This is quite important but not discussed here. Next, we focus on the study of ECG signals.

25.1 ECG Signals

The ECG signal consists of low amplitude voltages in the presence of high offsets and noise. One can categorize the ECG signals systematically in the following way. One finds two main categories: Bioelectric signals and Bio-impedance signals.

25.1.1 *Bioelectric Signals*

Biomedical engineering is an interdisciplinary area. It is concerned with the disciplines of development and manufacture of medical devices, diagnostic devices, drugs etc. All these require bioelectrical signals. These bioelectrical signals are typically very small in amplitude and amplification is required to accurately record, display and analyze the signals. The nerves and muscles create the signals. The analysis requires techniques from signal analysis in mathematics. The signals are often hidden in a background within other signals and noise components. They are

generated by dynamic biological processes with several parameters that are varying continuously.

The Electrocardiogram (ECG or *ecg*) measures the electrical conduction system of the heart. The ECG picks up electrical impulses generated by the polarization and depolarization of cardiac tissue and translates it into a waveform. In contrast to speech, the body has no intention to send a message that contains information to the receiver. Instead, the body contains information about its present state and part of the information is contained in the signals. The task is to obtain as much of this information as possible.

In this chapter, we will give a general overview of the ECG signals, ECG signal processing and the interpretation as well as classification of the signals for the diagnosis and for abnormality detection. In addition, we discuss how the signal is contaminated by other signals from other parts of the body, and how the noisy ECG signal is enhanced. We will discuss this with examples in this chapter. The discussion is limited to the human heart; we will not discuss animal's hearts.

The waveform is used to measure the rate and regularity of heartbeats, as well as the size and position of the chambers, the presence of any damage to the heart, and the effects of drugs or devices used to regulate the heart. The electrocardiography records the electrical activity of the heart captured over a period time. For this, external electrodes are attached to the surface of the skin and their recordings are captured on an external device. It turns out that the signal processes are of statistical character, i.e. one has to do with stochastic events. Here, one has to do with HSMs in a non-stationary situation.

Bio-impedance Signals

Measurement of the electrical impedance of tissue is used, for instance, in the diagnostics of the cardiovascular system. Another important application is the determination of measurements of the human body.

Most important measurable parts are:

- Total body water
- Fat free mass
- Lean body mass
- Fat mass
- Body cell mass
- Extracellular mass.

To determine those parameters, a precise measurement is required. The signals result from applications of the change in impedance versus the frequency used in the measurement of electrical streams. It measures the electric resistance. The impedance analysis uses methods for estimating body composition, in particular body fat. We model it as a resistor and capacitor. Bio-impedance signals often have to be acquired from a number of signal sources scattered over some area. This is often realized based on multiplexers. Digitizing of bio-impedance signals is based on an analog/digital converter. From these one can detect two quantities:

- The resistance R for analyzing the state of the liquid in the body.

- X_c : The sum of all membrane capacitance. It indicates the quantity of the body cell mass and quality of the body cells.

This illustrates how much information can be contained in a stochastic process.

25.1.2 Noise

Analog Front-End processing forms an important part of the ECG system since it needs to distinguish between noise and the desired signal that is of small amplitude. The main noise sources in ECG are:

- **Baseline wander**

The severity of the baseline wander is dependent on the HPF cut-on frequency and the PSD of the signal around DC. The output of the high-pass filter will have equal areas above and below zero volts. If the pulse repetition frequency is low compared with the time constant of the input circuit, we'll see some "droop" in the waveform, indicating the discharging of the input capacitor.

- **Power line interference**

The PLC signals travel through power lines. This makes reliable communication more difficult and is the cause of errors.

- **Noise from muscles**

The muscles have an influence on the body including the heart that have an effect on the signals.

In a conventional ECG, ten electrodes are placed on the patient's limbs and on the surface of the chest. The overall magnitude of the heart's electrical potential is then measured from twelve different angles ("leads") and is recorded over a period of time (usually 10 s). In this way, the overall magnitude and direction of the heart's electrical depolarization is captured at each moment throughout the cardiac cycle. During each heartbeat, a healthy heart will have an orderly progression of depolarization that starts with pacemaker cells in the sinoatrial node, spreads out through the atrium, passes through the atrioventricular node and then spreads throughout the ventricles. This orderly pattern of depolarization gives rise to the characteristic ECG tracing.

An ECG produces a pattern reflecting the electrical activity of the heart and usually requires a trained clinician to interpret. It can give information regarding the rhythm of the heart whether that impulse is conducted normally throughout the heart, or whether any part of the heart is contributing more or less than expected to the electrical activity of the heart.

The above mentioned sources have different influences on the types of the noise. There are different removal methods depending on the type of noise. Electrocardiography is the recording of the electrical activity of the heart and the measuring the heart's electrical conduction system. Electrical properties of tissues have been described since 1872. The general symptoms are studied including myocardial

infarction and pulmonary embolism. Modern ECG monitors offer multiple filters for signal processing.

25.2 EEG Signals

The electrocephalogram (EEG) is a recording of the electrical activity of the brain from the scalp. The recorded waveforms reflect the cortical activity. The EEG activity is quite small measured in microvolts (mV) and the main frequencies of the human EEG waves are delta, theta, alpha, and beta. The delta wave has the frequency of 3Hz, the theta has a frequency 3.5 to 7.5 Hz, alpha has a frequency between 7.5 to 13 Hz, and the beta wave has the frequency 14 and greater.

25.2.1 General Properties

The electrocephalogram (EEG) signal discusses the activity of the brain. It goes back to the second half of the nineteenth century. An EEG signal measures the current flow of the synaptic excitation's of the dendrites of the neurons in the cerebral cortex. It is generally performed at a frequency of 50 kHz. At this frequency, the current passes through both the intra-cellular and extra-cellular fluid and the total body water (TBW) may be calculated. Hence it is the recording the electrical activity along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain. It involves taking impedance measurements at less than seven frequencies.

Neurons pass signals via action potential created by exchange between sodium and potassium ions in and out of the cell. When the wave of ions reaches the electrodes on the scalp, they can push or pull electrons on the metal on the electrodes, the difference in push, or voltage, between any two electrodes can be measured by a voltmeter. Recording these voltages over time gives us the EEG as a stochastic process. Scalp EEG activity shows oscillations at a variety of frequencies. Several of these oscillations have characteristic frequency ranges, spatial distributions and are associated with different states of brain functioning.

Up to now, no systematic model is available that models precisely biomedical signals or their patterns. Therefore, one has to rely on experiments and observations and has to use techniques for these stochastic processes. Many neurons need to sum their activity in order to be detected by EEG electrodes. Synchronized neural activity produces signals. The brain's electrical charge is done by the Neurons that are electrically charged. Neurons are constantly exchanging ions with the extracellular milieu. This has to be performed in a regular way. It is a property of the signal process. To detect this property is a major challenge. There is also a possible feedback with explanatory character to the nodding disease. ECG heterogeneity is a measurement of the amount of variance between one ECG waveform and the

next. One can measure the heterogeneity by placing electrodes on the chest and then computing the variance of the signals obtained from these electrodes. In EEG there are generally typical difficulties discussed. Mainly they are the discovery of properties of the obtained signal processes.

25.2.2 Signal Types and Properties

The purpose of the ECG is to detect and amplify small electrical changes on the skin that are caused when the heart muscle depolarizes during each heartbeat.

The information source conducted by a nerve is called action potential (AC). The connection from the outside to the signal source is provided by the Brain-Computer Interface).

In case of the EEG signal, we discuss the diagnosis, monitoring, and treatment of abnormalities of the EEG signal by applying signal processing techniques and machine learning techniques. The electrical signal generated by a single neuron is very small and cannot be picked up by some EEG receiver. The action potential generates a wave. When the wave of ions reaches the electrodes on the scalp, they can push or pull electrons on the electrodes. This reflects the activities of many (thousands) of neurons in the brain. The measured data of both ECG and neurons explains that the relationship between the two is very complex. It can, however, be used for the following purposes:

- to identify epileptic seizures
- to identify organic encephalopathy or delirium
- to serve as an adjunct test of brain death
- to prognosticate in patients with coma
- to determine whether to wean anti-epileptic medications.

For these aspects different methods have been developed to reduce them to certain properties of signal processes. They often rely on prediction equations and algorithms to calculate results. These algorithms have in general been determined from healthy subjects. Filters are often used in ECG. Low-frequency filters are set at either 0.5 Hz or 1 Hz and the high-frequency is set at 40 Hz; both are used.

25.2.3 Disadvantages

There are not only advantages of EEG but also some weak points. EEG poorly measures some neural activities in the brain. It has a low spatial resolution and cannot identify certain regions in the brain. The signal-to-noise ratio is not very good and therefore one has to make many measurements. Next, we present some illustrations of sample recordings. One sees the results of measurements over time

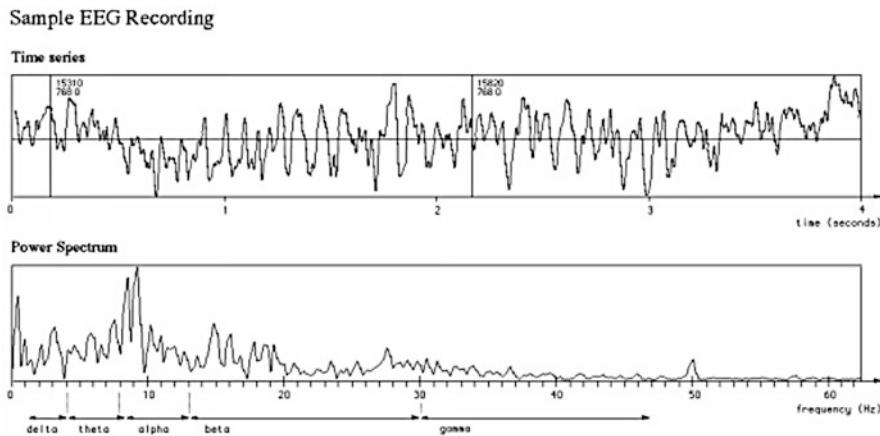


Fig. 25.2 Sample recording and power spectrum of EEG signal

from EEG. The samples range can go up to 20,000 Hz. This is illustrated in Fig. 25.2.

25.3 Neural Network Use

Since the EEG is a complicated and is a multicomponent process and no precisely known structure is available, it is usually difficult to decide what parameters should be chosen as a base for such classification. It then seems to be reasonable to use for this purpose the EEG features pertinent to certain psychic functions, which are considered as the point of interest in a given research. If so, it becomes necessary to have some means of relating EEG features to psychic processes, provided that no a priori knowledge on this item is available.

Epileptic seizures are diseases where neural networks are applied. The diseased tissue are manifestations of epilepsy. The detection of epileptic forms discharges in the EEG is an important component in the diagnosis of epilepsy. The detection of epileptic form discharges in the EEG is an important component in the diagnosis of epilepsy. As EEG signals are non-stationary, the conventional method of frequency analysis is not highly successful in diagnostic classification.

Wavelet transform is particularly effective for representing various aspects of non-stationary signals such as this where trends, discontinuities and repeated patterns of the other signal processing approaches fail or are not as effective. Through wavelet decomposition of the EEG records, transient features are accurately captured and localized in both time and frequency context.

Artificial neural network can deal with the analysis of EEG signals using wavelet transform and classification using and logistic regression. One can try to classify

EEG signals from healthy and unhealthy patients using a neural network. Feed forward nets with the back propagation method are also used.

In similar contexts neural nets are useful. As an example one can implement an adaptive feature model for the front-end feature extraction module in the form of a neural network, where the network weights represent the parameters of the model. The type of network used for the problems here are back propagation networks with the neurons as McCP ones. Because of the individual property of the measurements the parameters of the net are diverse and for each situation unknown. To obtain them one has to apply machine learning techniques. This is discussed in Part [II](#).

Medical areas that have seen the successful implementation of this technology include drug development, patient diagnosis, and image analysis. Two major cornerstones are the detection of coronary artery disease and the processing of EEG signals.

25.4 Major Research Questions

There are quite many research questions and some important ones are:

- (1) Can just one static sensor on the forehead make up for a grid of sensors placed across the scalp?
- (2) What kind of classification of brain waves will it be possible to make using a neural network? And if possible, then please elaborate on how stable is this classification?
- (3) What abstract information is contained in the output of a BCI?
- (4) How one can read the output of a BCI?

How satisfying or useful is a resulting and working BCI system is a basic practical problem.

25.5 Background Information

General and Past

For a general medical view on heart diseases see Braunwald ([1997](#)). The history of ecg goes back to the first half of the nineteenth century. The first practical ECG was invented by Willem Einthoven in the early 1900's. He was awarded the Nobel Prize in Medicine for his discovery in 1924. The German physiologist and psychiatrist Hans Berger (1873–1941) recorded the first human EEG in 1924. Most ECGs are performed for diagnostic or research purposes on human hearts, but may also be performed on animals, usually for diagnosis of heart abnormalities or research. Noisy ECG signals are discussed in Aditya et al. ([1996](#)). Adaptive digital notch filters for the elimination of power line noise from biomedical signals have been investigated in Ferdjallah and Barr ([1994](#)). Bio-impedance was first used over 30

years ago to measure the total water content of the body. EEG is most often used to diagnose epilepsy, which causes obvious abnormalities in EEG readings. It is also used to diagnose sleep disorders, coma, encephalopathies, and brain death. EEG used to be a first-line method of diagnosis for tumors, stroke and other focal brain disorders. In Saeid and Chambers (2007), one finds a broad overview over EEG computations. It is also used to assess patients with systemic disease, as well as monitoring during anesthesia and critically ill patients, see Lukaski et al. (1986).

Suggestion

In Amit et al. (2009), one finds applications of cluster analyzing the heart sounds. For the history of EEG see Electrocardiography (ecg, from Greek: kardia, meaning heart). For general EEG see Kyle et al. (2004). For recording of the activity of brain we recommend Swartz (1998). See EEG signals (EEG signals recorded from healthy volunteers with eyes open, epilepsy patients in the epileptogenic zone during a seizure-free interval, and epilepsy patients during epileptic seizures) were classified with the accuracy of 94.83% by the combined neural network. Also Niedermeyer and da Silva (2004) [Sw]. Notch filters (see part A) are discussed in Ferdjallah and Barr (1994). Neural networks have been applied frequently in these areas. To connect the nodes with network weights which specify the strength of the connection is a general task. In this context it has been done through a learning process. For using neural nets and fuzzy logic see Srinivas1 et al. (2013). Since the 1980ties HSM was used for Nukleotid- and Protein sequences and are since then standard in bioinformatics. An application of fuzzy sets for visual analysis of biomedical signals is described in Stuchlik et al. (1995). Neural network applications are described in Subasi and Erçelebi (2005).

Reference used in this chapter Moyer (2012) is recommended to readers for more information.

25.6 Exercises

Exercise 1 (For People with Medical Background) Look at patients with heart problems. Describe the uncertain health depending on heartbeat using the rough set method.

Exercise 2 Suppose you have a heart disease H for which you have fuzzy information coming from ECG. Describe the activities of a medical doctor using rough sets.

Exercise 3 Decompose an EEG signal into its basic frequency components.

Exercise 4 Describe a small vocabulary for giving commands for monitoring machines of your choice.

References

- [Braunwald1997]** Braunwald E. (Editor, 1997), Heart Disease: A Textbook of Cardiovascular Medicine, Fifth Edition, Philadelphia, W.B. Saunders Co.
- [Swartz1998]** Swartz, Barbara E. (1998): The advantages of digital over analog recording techniques. *Electroencephalography and Clinical Neurophysiology* 106 (2): 113–7.
- [Niedermeyer2004]** Niedermeyer E. and da Silva F.L. (2004). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincot Williams & Wilkins.
- [Ferdjallah1994]** Ferdjallah, M., Barr, R. (1994). Signals. *Biomedical Engineering*, IEEE.
- [Sanei2007]** Saeid Sanei, Chambers J.A. (2007) *EEG Signal Processing*, pp.312, Wiley.
- [Lukashi1986]** Lukaski HC, Bolonchuk WW, Hall CB, Siders WA (1986). Validation of tetrapolar bioelectrical impedance method to assess human body composition *J. Appl. Physiol.* 60.
- [Amit2009]** Amit, G., Gavriely, N. Intrator, N. (2009) Cluster analysis and classification of heart sounds. School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel Rappaport Faculty of Medicine, Technion-Israel Institute of Technology, Haifa, Israel.
- [Aditya1996]** Aditya,S.K, Chu,C.-H., Szu, H.H. (1996). Application of adaptive sub band coding for noisy bandlimited ECG signal processing SPIE 2762, Wavelet Applications III, 1996.
- [Moyer2012]** Moyer, V.A. (2012) “Screening for coronary heart disease with electrocardiography: U.S. Preventive Services Task Force recommendation statement.”. *Annals of Internal Medicine* 157 (7).
- [Srinivas2013]** Srinivas1, N., Babu, A. V, Rajak, M. D. (2013). Condition Monitoring and Analysis for ECG Signal Using Fuzzy Advance Neural Networks and Hypertext Preprocessor. *ijetae.com*. Certified Journal, Vol.3, Issue 4.
- [Kyle2004]** Kyle UG, et al. (2004) ESPEN Guidelines. Bioelectrical impedance analysis - part 1: review of principles and methods. *Clin. Nutr.* 23.
- [Stuchlik1995]** Pieter J, Stuchlik F, von Specht H, Mühler R. (1995). Fuzzy sets for feature identification in biomedical signals with self-assessment of reliability: an adaptable algorithm modeling human procedure in BAEP analysis. *Comput.Biomed.Res.*
- [Subasi2005]** Subasi A., Erçelebi, E. (2005). Classification of EEG signals using neural network and logistic regression. *Computer Methods and Programs in Biomedicine*, Vol. 118.

Chapter 26

Seismic Signal



Overview

The main principle of a seismic event is that it sends a vibration in the form of an elastic wave. This leads to the model using signal processes. Similar to the signal processes in medicine the wave does not have a precise semantics understandable to humans.

The applications include a reservoir of fluid movement monitoring, hydraulic fracture monitoring, and the prediction and analysis of earthquakes. Sensor arrays are placed to record seismic data, and the data are processed to detect and localize microseismic events that are buried in the noisy recordings. Typical microseismic processing techniques include signal-enhancing filters, signal detection, and array signal processing. One wants to derive from the analysis of seismic records information about the structure and physical properties of the Earth through which the seismic waves propagate as well as about the geometry, kinematics and dynamics of the seismic source process. This task is complicated by the fact that the seismic signals radiated by the source are weakened and distorted by geometric spreading and attenuation and, due to reflection, diffraction, mode conversion and interference during their travel through the Earth. Additionally, seismic signals are superposed and sometimes completely masked by seismic noise. This is in particular the case for low (SNR), e.g., relatively weak events.

A seismic source generates the seismic signals. A seismic source is a device that controls the performance of the reflection and refraction of the seismic energy. Examples of seismic sources are dynamite and air gun. Seismic sources can provide single pulses or continuous sweeps of energy. The area of seismic signal related studies are termed seismology. The solution to the inverse problem is commonly used to solve the seismology typed problems. In general, seismic processes are Hidden Stochastic Models because one does not know how precisely the sources operate. This reminds us to the psychoacoustic phenomena in speech. Seismic interpretation estimates geometry and properties the Earth. Again, this has a highly non-unique solution, and interpreters should always be advised to present it as such.

The main purpose of seismic surveys is to accurately record ground motion caused by known sources in a known locations. The record of ground motion with time constitutes a seismogram. The origin of seismic data is signal processes that are stochastic processes. This implies that the signal processes obtained are stochastic too. The other mentioned kinds of information refer to properties of these processes. A main problem is that data treated as noise in one context may be considered as useful signals in other applications. In this chapter, we will discuss this issue just in an abstract manner applying signal processing directions and machine learning tools.

26.1 Generalities

A seismic source signal has the following characteristics:

- An impulsive source generates it.
- It is band-limited.
- The generated waves are time-varying.
- Signal processes may last for a long time because the earth is moving this way.

Equation 26.1 describes these signals:

$$s(t) = \beta \exp^{\alpha t^2} \sin(2\pi f_{max} t) \quad (26.1)$$

In this equation, f_{max} is the maximum frequency component of the generated wave form.

26.2 Sources of Seismic Signals

There are different sources for seismic signals. Sometimes this takes place slowly and over long distances. Seismic sources can provide single pulses or continuous sweeps of energy. Both types of seismic sources generate seismic waves. One has to convert these waves to signals, the same as for speech. The main sources are:

- Earthquakes
- Volcanoes
- Explosions (especially nuclear bombs)
- Wind
- Vehicles
- People

Many methods have been developed for the seismic signals. The arguments of these methods are, however, not directly the signals. They are some higher order

mathematical concepts, mostly coming from physics. As a consequence, many of them can be measured. The signals are also distorted by the seismograph. The mechanical seismograph is a second-order high-pass filter. To know about the sources is important because of possible dangers as for instance from volcanoes. Often there are seismic impulses. The interior of the Earth at multiple scales using natural or artificial (e.g., urban) background noise as a seismic source can be explored. For example, under ideal conditions of uniform seismic illumination, the correlation of the signals between two seismographs provides an estimate of the bidirectional seismic impulse response. Explosives, such as dynamite, can be used as crude but effective sources of seismic energy. A spark gap sound source is a means of making very low frequency sonar pulse underwater. A Seismic vibrator propagates energy signals into the Earth over an extended period of time as opposed to the near instantaneous energy provided by impulsive sources. There are more methods of this kind that are artificially created and measures properties of the material through which they travel. The difference between seismic sounds and noise is not always quite clear.

26.3 Intermediate Elements

In the seismic area intermediate concepts are the concepts between the original seismic processes of interest to humans. We indicated this above. Energy is such an example. Others are:

- Gravimetry (measuring the changes in the gravity of the earth)
- Wavelet decomposition and processing
- Transportation of fluid and heat

These concepts usually split into different sub concepts. Their analysis is mathematically quite involved and their measurement is sophisticated.

26.4 Practical Data Sources

The data describing the reservoir characteristics include core data, well logs, well tests, production data and seismic surveys. Well logging, also known as borehole logging is the practice of making a detailed record (a well log) of the geologic formations penetrated by a borehole. Among the data, core data provide the most direct and accurate information. Well logs provide valuable but indirect information about mineralogy, texture, sedimentary structures and fluid content of a reservoir. Generally, well logs embody continuous information with high vertical resolution. In addition, compared to other data, well logs are easy to acquire in practice.

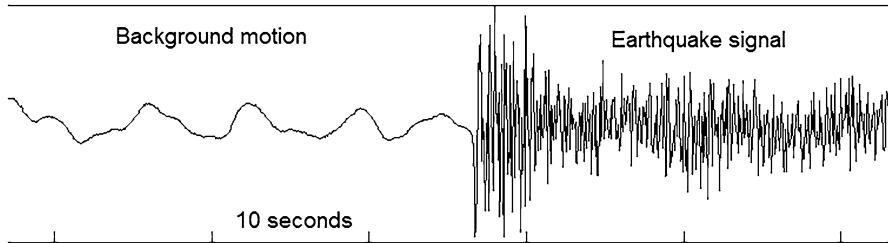


Fig. 26.1 Signal produced by earthquake

Example

Figure 26.1 is a seismogram from an event in Venezuela. At the left part of the seismogram one sees the natural background motion of the earth and to the right the earthquake signal. The signal amplitude is proportional to ground velocity.

26.5 Major Seismic Problems

Most of the problems in this area have relations to uncertainty. This starts when one considers the sources. A particular problem concerns seismic attenuation: the loss of energy as a seismic wave propagates through the earth. Many physical processes can lead to the attenuation of a seismic trace. They range from spherical divergence or scattering to intrinsic attenuation effects. Among the many types of seismic waves, one can make a broad distinction between *body waves* and *surface waves*.

- Body waves travel through the interior of the Earth, surface waves travel across the surface.
- Surface waves decay more slowly with distance than do body waves, which travel in three dimensions.

The ability of a material to attenuate seismic waves is measured by a dimensionless quantity Q , the attenuation factor, by Eq. 26.2.

$$Q = \frac{\text{Wave energy}}{\text{Dissipated energy per cycle}} \quad (26.2)$$

The attenuation of the wave has a direct link to the different layers that compose the Earth. Transient signals such as earthquake seismograms may have a complicated waveform, but this waveform is determined by the earthquake source, the structure of the Earth, and the properties of the seismograph. The amplitudes and phases of the harmonic components of the signal are not random numbers. They

are, in principle, smooth functions of frequency, often smoother and simpler than the signal appears in time domain.

26.6 Noise

The noise can be:

- External noise
- Intrinsic noise

The intrinsic noise is caused by environment. There are two principal noise types:

- Time varying noise
- Strong noise

The intrinsic noises come from the earth and the measurement itself. Some are mentioned below. We treated the types of noise above. For the seismic signals it is however useful to discuss the sources of the noise. This is particular difficult because the signal goes through the earth that is just partially known and has various not precisely known influences. This depends to a large degree on the location where the measurement is taken; for instance on the distance to an ocean. The intrinsic noise comes from elements as residual noise coming from the measurement. The sources of noises contain information about their possible locations. The remaining difficulty is to determine the type of noise. The disturbance of noise is very different from most as for instance the medical measurements.

The main sources of noises:

- Ambient vibrations due to natural sources (like ocean microseisms, wind, etc.);
- Man-made vibrations (from industry, traffic, etc.);
- Secondary signals resulting from wave propagation in an inhomogeneous medium (scattering);
- Effects of gravity (like Newtonian attraction of atmosphere, horizontal accelerations due to surface tilt);
- Signals resulting from the sensitivity of seismometers to ambient conditions (like temperature, air pressure, magnetic field, etc.);
- Signals due to technical imperfections or deterioration of the sensor (corrosion, leak-age currents, defective semiconductors, etc.);
- Artifacts from data processing.

In most cases, there is no strong noise. Sometimes one does not want to know the exact strength of the noise. This is for instance the case if one wants to detect a specific earthquake signal. For this, some qualitative information is sufficient and some small or short noise is not disturbing. On the other hand one has to listen for a longer time and handling of signals for a longer time is sometimes more difficult.

26.7 Background Information

Seismic signals have been of interest to humans for centuries, mainly for the forecast of earthquakes. Earthquakes do not occur very often but they are very expensive. In modern times early seismic exploration is based on refracted waves (1919–1921). A problem is to detect the signs of an earthquake quite early. The commercial use came much later. From the 1930s, reflected waves started to use for oil exploration. Modern seismic explorations are primarily based on the reflected waves and of commercial interest. Besides oil, there are many other valuable minerals that can be detected from listening for seismic signals. However, there are some differences to the other applications in the book.

Seismic signals depend not only on time but also on the location where they are recorded. It is therefore meaningful to ask whether the signal recorded at one location is coherent with the same signal recorded at another location. Sometimes it is not necessary to know exactly the amplitudes of the signals, and small noises can be ignored. In most of the stations, the signal is too weak for direct observation on the seismograms. One sees it most clearly on the spectra. In most of the stations the signal can be observed along the spectral profile. A good overview and many explanations about the topic is in Bormann and Wieland (2013). It gives also a good introduction to signal processes.

Another good source is Crawford et al. (1960). There is also a kind of close relation of the subject to speech recognition. In Goldstein and Archuleta (1987), there is a method for analyzing seismic signals recorded at an array of seismometers. Material about Eq. 26.1 can be found in Boarding and Lines (1997). It is of general interest to seismic modeling.

References used in this chapter such as Shearer (2009), Sheriff (1991), Bauer et al. (2014), Goldstein and Archuleta (1987), Shapiro et al. (2005), Sengel (1981), Chouet (1996), Benítez et al. (2007), Cortés (2007), and Havskov and Ottemöller (2009) are recommended to the readers.

26.8 Exercises

Exercise 1 A result often used in the study of discrete (time random signals is) the following summation formula:

$$\sum_{t=1}^N \sum_{s=1}^N f(t-s) = \sum_{\tau=-N+1}^{N-1} (N - |\tau|) f(\tau)$$

where f is an arbitrary function. Give a proof of the formula.

Exercise 2 A source sends out a signal $s(t)$. What is the spectrum of the signal? What is its use?

Exercise 3 Give a formal description of ambient vibrations due to natural sources that have been introduced above: (like ocean micro-seisms, wind, etc.)?

References

- [Bormann2013]** Bormann, P, Wieland, E. (2013). Seismic Signals and Noise. In Bormann, P. (Ed.), New Manual of Seismological Observatory Practice 2 (NMSOP2), Potsdam: Deutsches GeoForschungsZentrum GFZ
- [Crawford1960]** Crawford, J. M., Doty, W. E. N. Lee, M. R., (1960). Continuous signal seismograph: Geophysics, Society of Exploration Geophysicists, 25.
- [Boating1997]** Boarding Phil, Lines, R.L. (1997): Seismic Wave Propagation Modeling and Inversion Society of Exploration Geophysicists, Seventh Printing 2008.
- [Sheriff1991]** Sheriff R. E. (1991): Encyclopedic Dictionary of Exploration Geophysics, Society of Exploration Geophysicists.
- [Bauer2014]** Bauer, M, Freedon. W., Jacobi, H., Neu, T., (2014). Handbuch Tiefe Geothermie. Springer Verlag.
- [Goldstein1987]** Goldstein P, Archuleta, R.J. (1987) Array analysis of seismic signals. Geophysical Research letters 14.
- [Shear2009]** Shearer, P.M. (2009). Introduction to Seismology. Cambridge University Press.
- [Shapiro2005]** Shapiro,N.M., M., Stehly,L., Michael H. Ritzwoller,M.H. (2005). High-Resolution Surface-Wave Tomography from Ambient Seismic Noise. Science 11, 2005.
- [Sengel1981]** Sengel, E.W. (1981). Handbook on well logging. Oklahoma City, Oklahoma: Institute for Energy Development.
- [Chouet1996]** Chouet, B., (1996). Monitoring and mitigation of volcano hazards, chap. New methods and future trends in seismological volcano monitoring ed , Scarpa R.,Tilling R pp. 23–97, Springer-Verlag.
- [Benítez2007]** Benítez,C., Ramírez,J., Segura,J.C., Ibáñez,J.M., Almedros,J., García-Yeguas, A.
- [Cortés 2007]** Cortés, G. (2007). Continuous HSM-based seismic event classification at Deception Island. IEEE Trans. Geoscience and Remote Sensing, vol. 45.
- [Havskov2009]** Havskov, J., Ottemöller, L (2009). Processing Earthquake Data. <ftp://geo.uib.no/pub/seismo/>

Chapter 27

Radar Signals



Overview

Radar transmits electromagnetic waves to detect and locate objects. The analysis of the radiation energy into space, echo reflection, detection of objects, tracking determination of the distance, speed are discussed in radar signal processing. The clutter from the earth surface can distort the target echoes. Radar is applied in both civilian and military applications. Moving target detection (MTI) is an application of radar signal processing. Such a detection is possible, when the receive signal is adequately processed. Radar tracking is another important radar signal processing application. Tracking is a process of determining the speed, and direction of targets which enables monitoring the target throughout the radar cover area. Tracking can be performed successfully, if detection are performed successfully. Searching, and detecting, tracking and imaging are some radar applications. A very brief discussion about the radar signal processing as an essence of an application of signal processing is provided in this chapter.

27.1 Introduction

Transmitter, receiver, antenna and target are some principle radar components. Radar transmits an electromagnetic wave, and the receiver receives the returned echo to search for targets. Transmitter amplifies an radio frequency (RF) carrier modulated signal. Amplifier, antennas and waveform generators are typical transmitter components. Transmitter radiates electromagnetic energy. Radiated electric field is a function of the angle measured from the center of the beam. The receiver consists of RF filter, amplifier, mixers, and intermediate frequency (IF) amplifiers. The power is computed from processing antenna received signal.

If the radar transmitted signal denoted by $x_d(t)$ formulated in Eq. 27.1, and then the received signal is $x_r(t)$; these are formulated in Eq. 27.2.

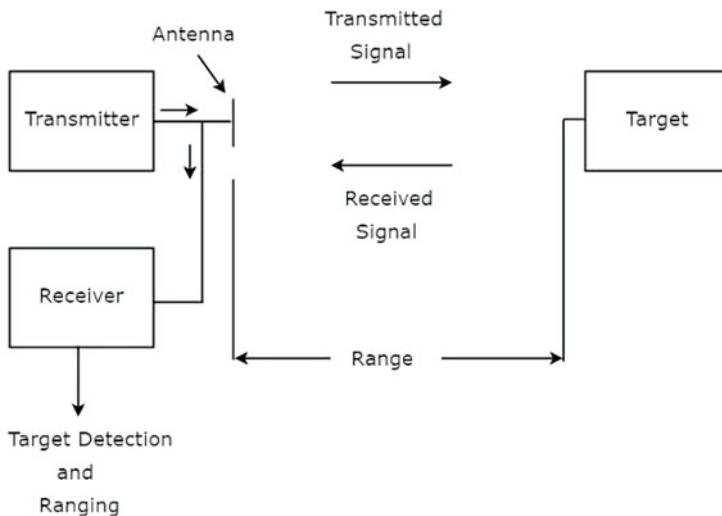


Fig. 27.1 Basic components of a radar system {https://www.tutorialspoint.com/radar_systems/radar_systems_overview.htm}

$$x_d(t) = x(t)e^{j\omega_c t} \quad (27.1)$$

$$x_r(t) = x(t - \tau)e^{j\omega_c(t-\tau)} \quad (27.2)$$

Equations 27.1 can be seen as an ideal mathematical model for the direct signal where $x_d(t)$ is the modulated information time series and $\omega_c = 2\pi f_c$ is the carrier frequency in cycles/sec. Then the reflected signal is $x_r(t)$ in Eq. 27.2.

Examples of direct and target signals in time and frequency domain are shown in Fig. 27.7.

In Fig. 27.1, an electromagnetic wave generated by the transmitter unit is transmitted by an antenna and the reflected wave from the objects or echo is received by the same antenna.

The reflectivity is the energy returned from an object. The reflection depends on the size, shape, and composition of the object. Some part of the reflected wave is received by the receiver antenna applying some complex signal processing, communication methods, and techniques.

Based on the transmitter and receiver position, radars can be mono-static, bi-static or multi-static.

One fundamental challenge in radar is detecting target signal reliably, and unambiguously that is itself obscured by interference from other sources. Detection of signals in the presence of noise, using classical Bayes or Neyman-Pearson decision criteria, is based on hypothesis testing.

27.2 Radar Types and Applications

In monostatic radar, the radar transmitter and the receiver antenna is at the same location as shown left in Fig. 27.2. Here the transmitter antenna first transmits the signal to the target and the receiver antenna reflects the received signal from the target. In bistatic radar, the transmitter and receiver are separated shown right in Fig. 27.2. Bistatic radars can operate with their own dedicated transmitters or with transmitters of opportunity.

The distance between the transmitter and receiver is the order of the expected distance to the target. When the transmitter of opportunity is a non-radar transmission, such as, communications, broadcast stations or global position satellites, then they are known as passive radar or passive coherent location or passive bi-static radar. The passive radar system is typically used for target detection, positioning and tracking passive coherent location. The passive radar uses illuminations by transmitters of opportunity, such as digital audio broadcasts (DAB), Sirius XM signal to detect and track targets. The SiriusXM radio signal is easy to set up for experiment than other different frequency band. The SiriusXM system is satellite based, terrestrial repeater enhanced, and digital broadcasting system.

Passive Bistatic Radar (PBR) makes use of emissions from broadcast, communications or radio navigation transmitters rather than a typical radar transmitter. The receiver is passive and potentially undetectable. The PBR is used in surveillance, atmospheric studies, ionospheric studies, oceanography, mapping lightning channels in thunderstorms and monitoring, radioactive pollution, target tracking and target classification.

The PBR is a special form of bistatic radar which makes separate of existing transmitters, such as broadcasting signal. The PBR is used in the military for various purposes such as moving target detection. The PBR does not transmit energy, and they are so far non-detectable. The PBR is useful, when the energy reflected by the target is very low.

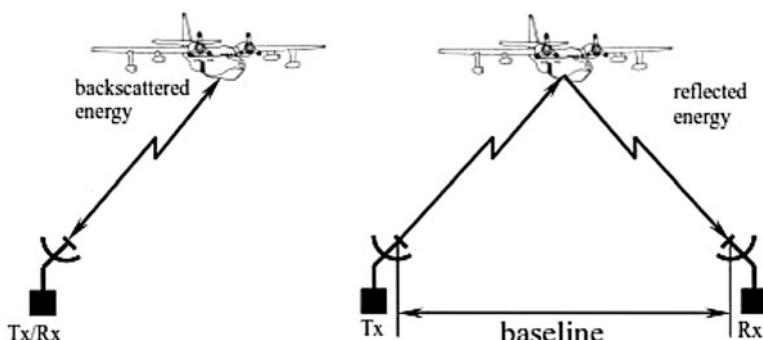


Fig. 27.2 Monostatic radar and bistatic radar <http://www.telecomabc.com/b/bistatic-radar.html>

27.3 Doppler Equations, Ambiguity Function (AF) and Matched Filter

“The Doppler effect or Doppler shift is the change in frequency of a wave in relation to an observer who is moving relative to the wave source” [Wikipedia]. In Eq. 27.5, c is the speed of light, λ_{rs} is the wavelength which is measured if the source is at rest and v_{rd} is the speed of the source moving along the line of sight. If the target moves at an angle with respect to the line of sight, then the Doppler shift $\Delta\lambda$ in Eq. 27.5 only informs the part of motion along the line of sight.

$$\Delta\lambda = \frac{\lambda_{rs} v_{rd}}{c} \quad (27.3)$$

Doppler radar sends a beam of electromagnetic radiation waves tuned at needed frequency to a moving object. This can be seen in Fig. 27.3 [Overview of radars. S. Cruz-Pol INEL 6069.]

The ambiguity function is the time response of a filter matched to a given finite energy of the signal, when the signal is received with a delay τ and a Doppler shift v relative to the nominal values by the filter shown in Eq. 27.4.

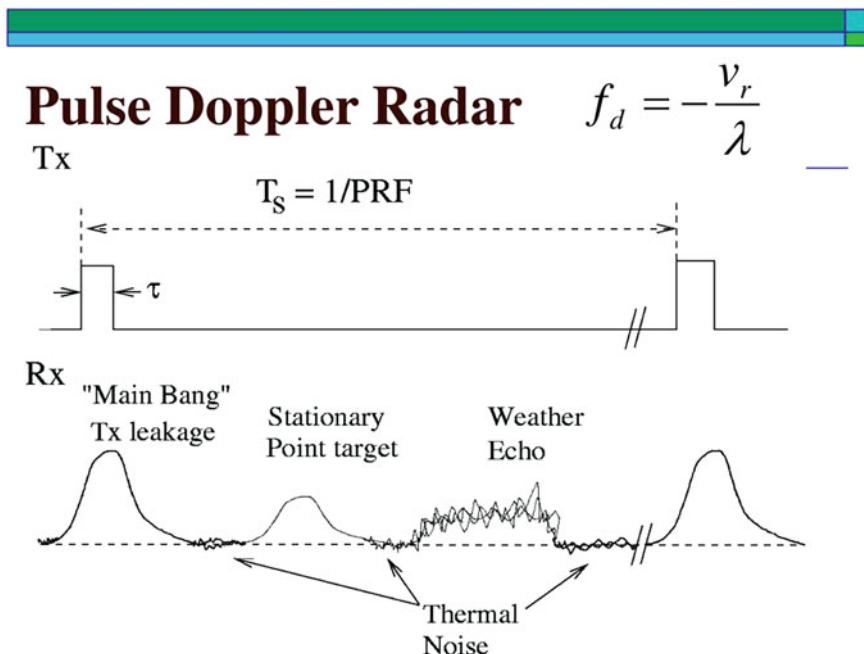


Fig. 27.3 Basic pulse Doppler radar overview {<http://doi.org/Overviewofradars.S.Cruz-PolINEL6069>}

$$|\chi(\tau, \nu)| = \left| \int_{-\infty}^{\infty} \mathbf{u}(t) \mathbf{u}^*(t + \tau) \exp(j2\pi\nu t) dt \right| \quad (27.4)$$

The ambiguity function is used in radar systems to get the distance and the relative speed of a moving object with respect to the transmitter. It is called ambiguity function because it tells about the ability to distinguish objects that are close between the and with a similar vector.

The ambiguity function in radar signal processing gives an understanding about the response of a signal processor to a given returned signal.

The frequency and wavelength of the electromagnetic waves are affected by relative motion. This is called the Doppler effect. In Eq. 27.5, c is the speed of light, λ_{rs} is the wavelength. If the source is at rest and v_{rd} is the speed of the source moving along the line of sight, the target moves at an angle with respect to the line of sight, then the Doppler shift $\Delta\lambda$ in Eq. 27.5 only informs the part of motion along the line of sight.

$$\Delta\lambda = \frac{\lambda_{rs} v_{rd}}{c} \quad (27.5)$$

A receiver bandwidth (BW) is assumed to be equal to that of the transmitted pulse. As a receiver, the matched filtering is done separately on the returns from each pulse, after which the signal is sampled by A/D converter and sent to a digital processor. The digital signal processing performs all subsequent radar signal and data processing.

In Eq. 27.6, c is the speed of light, ρ is range resolution, τ is pulse duration, B is signal bandwidth.

$$\rho = \frac{c\tau}{2} = \frac{c}{2B} \quad (27.6)$$

Matched filter maximizes the SNR in the received signal. The response of the matched filter is described by the auto-correlation of the signal is in Eq. 27.7.

$$\mathbf{s}_0(t) = \int_{-\infty}^{\infty} \mathbf{s}(\tau) \mathbf{s}^*(\tau - t) d\tau \quad (27.7)$$

27.4 Moving Target Detection

The frequency and wavelength of electromagnetic waves are affected by relative motion. The relative motion between radar and target introduces Doppler effect. The approaching or preceding component of motion produces Doppler effect. As the source of the electromagnetic wave progresses, the frequency increases and the wavelength decreases. If the source moves back, the frequency decreases and the wavelength increases. The motion of the object causes a wavelength shift $\Delta\lambda$ based

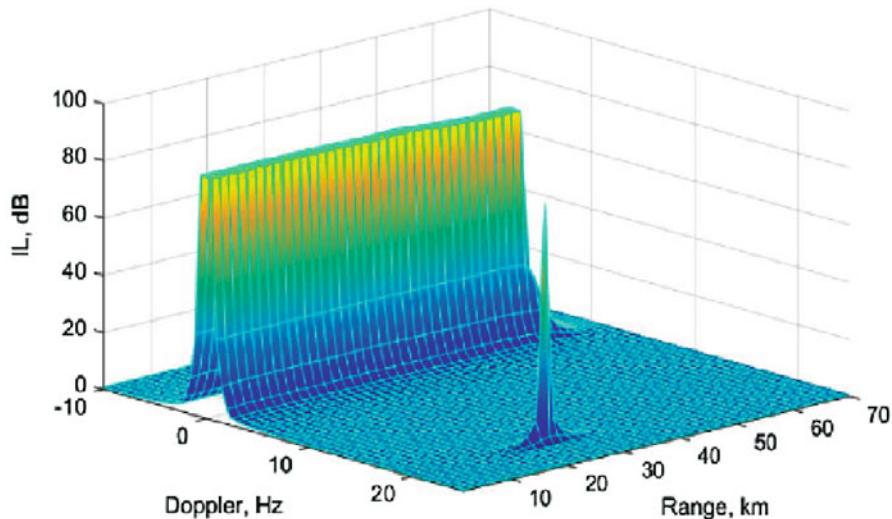


Fig. 27.4 Moving target detection <http://www.jpier.org/PIERM/pier.php?paper=16122501>

on the speed and direction of the moving object. The amount of shift depends on the source's speed and it is given by Eq. 27.8.

$$\Delta\lambda = \frac{2R}{c} \quad (27.8)$$

Pulse radar systems use special filters to distinguish between slow moving or stationary targets and fast-moving targets. Such filters are known as Moving Target Indicator (MTI). The MTI filter has a deep stopband at DC and at integer multiples of pulse radar frequency that are used to suppress the clutter returns little or without any degradation. In MTI, the Doppler effect assumes stationary objects as clutters with zero frequency, and the transmitted waveform frequency is filtered out from the received signal. Aircraft, ship, missile, bird, people, natural creatures are some examples of target. The target and the clutter are separated by Doppler shift. The Doppler shift is measured by Eq. 27.9 where f is the transmission frequency, c is the speed of light v_r is the propagation velocity, \dot{R} is the range rate (Fig. 27.4).

$$f_d = \frac{2\dot{R}c}{f} = 2\frac{v_r}{\gamma} \quad (27.9)$$

27.5 Applications and Discussions

Radar has a wide range of applications. Police traffic for enforcing speed limits, weather radar, military applications for surveillance navigation and weapons guidance for ground, sea, air, and space vehicles are some typical examples. Radar is also used for collision avoidance and buoy detection by ships, automobile, and trucking industries. The space borne, satellite and space shuttle, airborne radar are used in mapping, earth topology and environmental characteristics such as water and ice conditions, forestry conditions, land usage, and pollution. Detection of an object/target, and identify the object is a typical radar signal analysis problem. This requires determining receiver's output at a given time which contains the information of the measurement the echo from a reflecting object. The decision on detection is based on the amplitude $A(t)$ of the receiver output (where t represents the time) to a threshold $T(t)$, which may be set apriori in the radar design or may be computed adaptively from the radar data. The time required for a pulse to propagate a distance total $2R$ and return is $\frac{2R}{c}$; if $A(t) \geq T(t)$ at some time delay t_0 after the pulse is transmitted, and the target is at range $R = \frac{ct_0}{2}$, c is the speed of light. Once the object has been detected, the next step is to tract the location, and velocity of the object. Velocity is estimated by measuring the Doppler shift of the target echoes. Doppler shift provides only the radial velocity component, but a series of measurements of position and radial velocity component can be used to predict the target dynamics in all directions. Strategic and tactical surveillance, remote atmospheric and sea state sensing, tracking and guidance, and precision disaster control or monitoring are examples of radar signal applications. In Doppler effect, the stationary objects do not have frequency shifts, the transmitted frequency is filtered out the received signal. A challenge in signal detection is to detect reliable, unambiguous target signal that is itself obscured by interference from other sources. Detection, tracking, moving target detection (MTI) are some typical radar signal analysis applications. Clutter refers to returns from the earth surface, electromagnetic interference, meteor, lightning, and from the vicinity of other targets of interest.

27.6 Examples

Example 27.1 Pattern recognition is used in radar to detect a presence of certain signal. For example when an aircraft takes signal reflected to it, the reflected signal can be analyzed to find the presence of the target signal and disguise the targeted real input. A signal detector detects the presence of the object (George 2007).

Example 27.2 Unwanted radio wave reflections in radar systems are called clutter. The clutter can obscure desired targets. Reflections from the ground, sea, weather, mountains and birds can be some examples of clutter. In weather radars, the reflections from storm systems are desired, and reflections from airplanes are clutter.

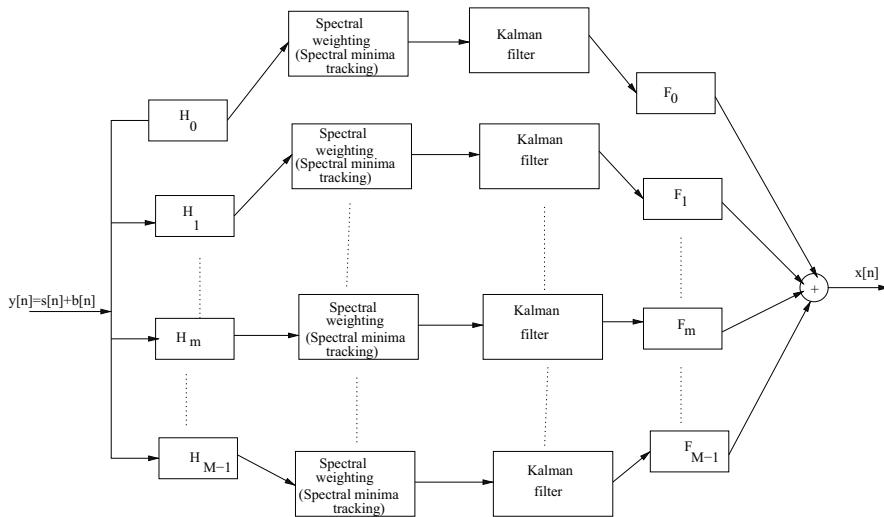


Fig. 27.5 Adaptive spectral M-subband Kalman filter in passive radar signal analysis

The clutter is often unwanted. An example of adaptive M-band Kalman filter in bistatic radar signal analysis is presented here. This approaches Bayesian approach.

The adopted approaches in this example discussed in Chaps. 6 and 7 are in Fig. 27.5:

- Sub-band decomposition
- Noise tracking in the sub-band by spectral noise minimization
- Colored Noise Kalman filtering

S-band i.e., in 2 to 4 gigahertz (GHz), 15 to 7.5 cm wavelength range signal in time and frequency domain is shown in Figs. 27.6, 27.7, 27.8 and 27.9.

The zero-Doppler cut of the AF is the autocorrelation of the pulse and it is computed by Eq. 27.10 and shown in Fig. 27.10.

$$A(\tau, 0) = |x(t)x^*(t + \tau)| \quad (27.10)$$

The enhanced signal is shown in Fig. 27.11.

27.7 Background Information

Radar is a detection system that has been used in military and civilian applications to detect aircraft, ships, motor vehicles and weather tracking and urban areas. Radio waves are part of electromagnetic spectrum. Radars use radio-frequency transmissions. The first radar was issued in Germany in 1904. Range and sensitivity

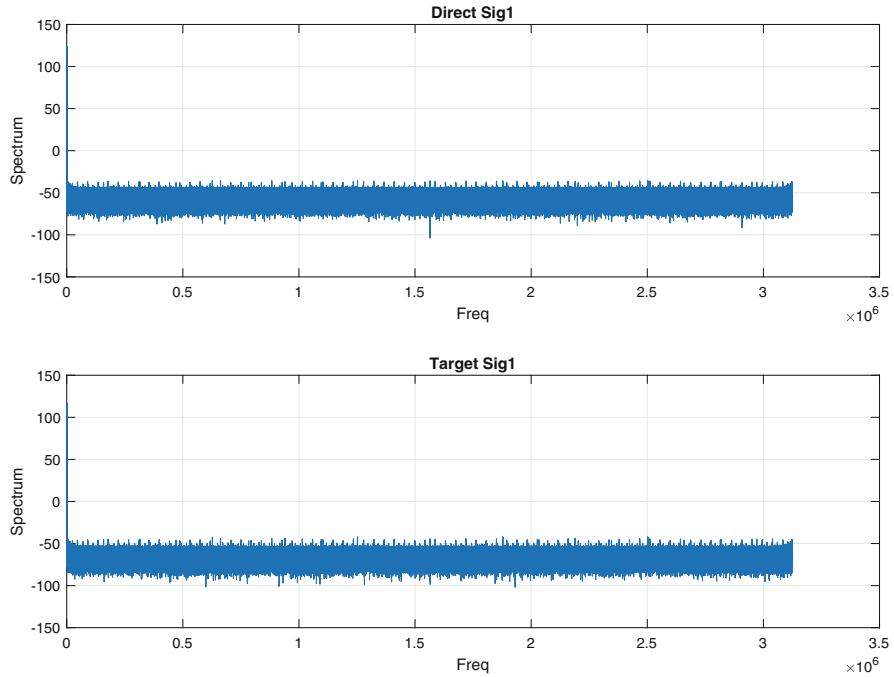


Fig. 27.6 S-band signal in time domain

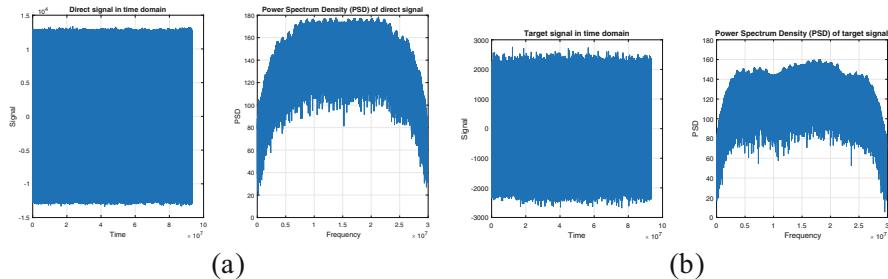


Fig. 27.7 (a) Direct and (b) Target signal in time and frequency domain

of radars were constantly improved since then. The microwave radar has been built. The progress and applications of radars have been continued.

There are different types of radar systems used for different applications: police enforcement for traffic control, space exploitation using RADAR SAT, remote sensing using SATELLITE, moving target detection.

Radar was initially invented in 1930 for military applications because of the need for air defence operating land, sea, and air. In the battlefield, radar is used for the surveillance over the sea, air, reconnaissance targeting land, sea control weapons, locating hostile weapons, detecting intruders. This chapter gives a very brief outline

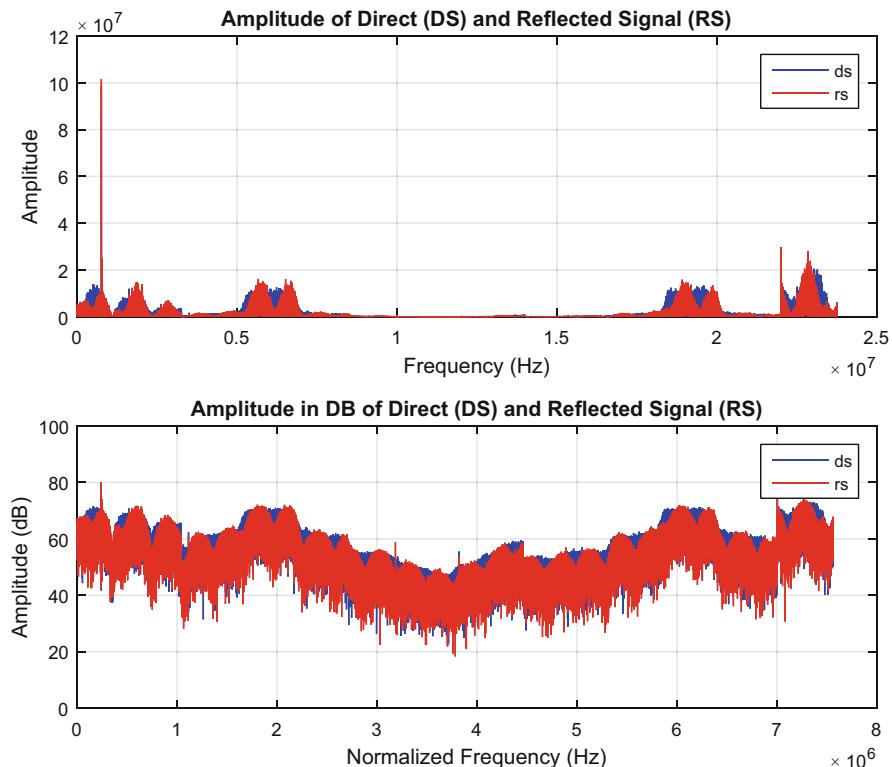


Fig. 27.8 Spectral analysis of S-band signal for passive radar

of radar signal analysis. Handbook by NAWCWD TP 8347 Fourth Edition 2013, Principles of Modern radar, Volume 3: radar Applications by William L. Melvin, James A. Scheer. The history of radar signal is long and extensive. This topic has numerous reports and lecture series.

References such as Mahafza (2008), Budge and German (2015), Kange and Eyunig (2008), Kolawole (2002), Skolnik (2008), Li et al. (2005), Zamani and Abbas Sheikhi (2017), Theodoridis and Chellappa (2014), and Pouliarikas and Dorf (2018) are used in this chapter. Readers are encouraged to read these or similar references for details.

27.8 Exercises

Question 27.1 Pulse radar transmits a peak power of 1 MW. Its pulse repetition rate is 100 ms, and the transmitted pulse width is 1 ms. Calculate: maximum unambiguous range, average power, duty cycle, transmitted signal energy, and bandwidth.

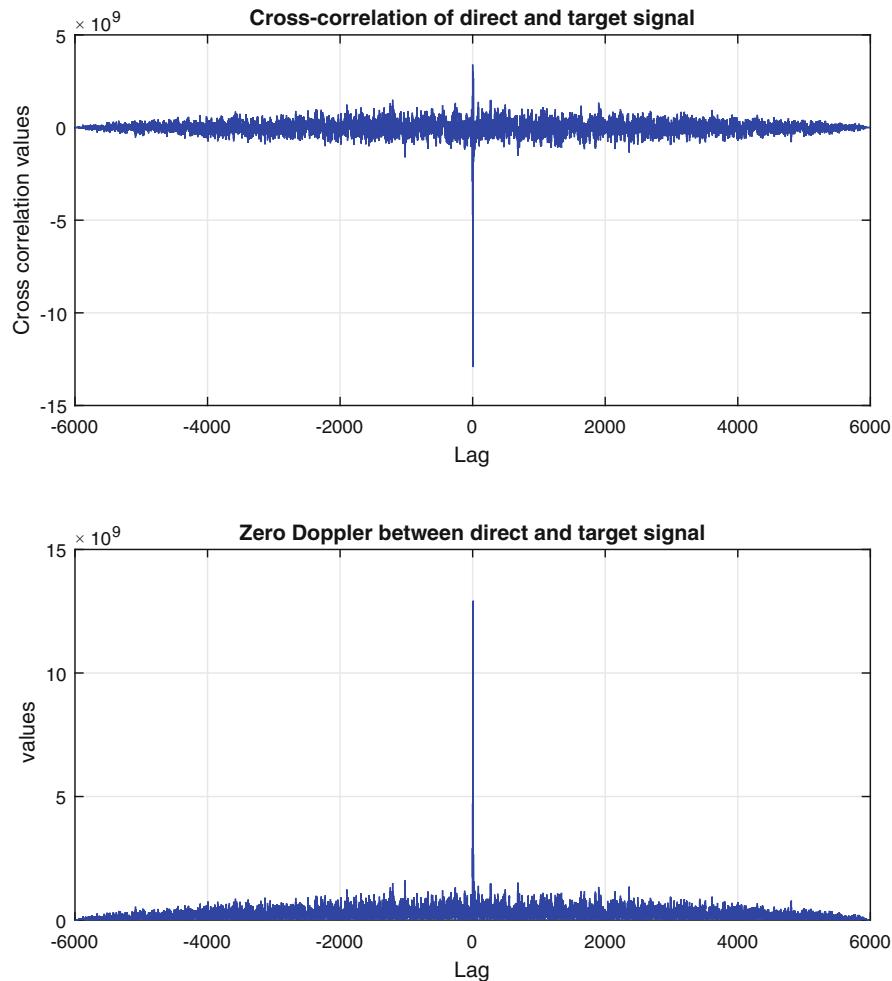


Fig. 27.9 Cross-correlation of direct and target signal

Question 27.2 How spectral estimation can be applied to improve the radar signal processing range-angle estimation accuracy? Give an example!

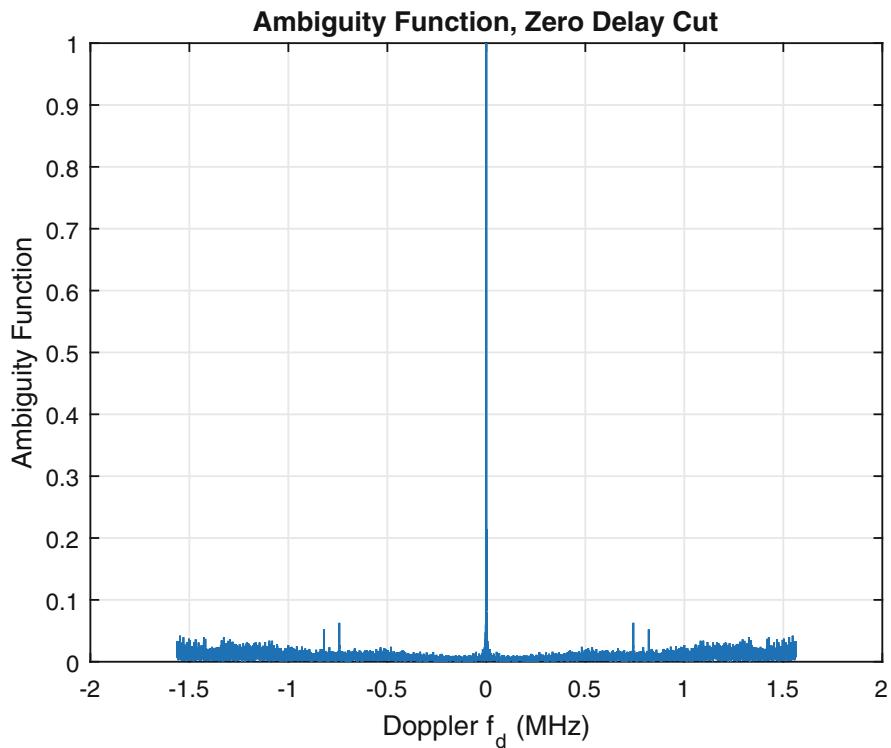


Fig. 27.10 Ambiguity function at zero delay

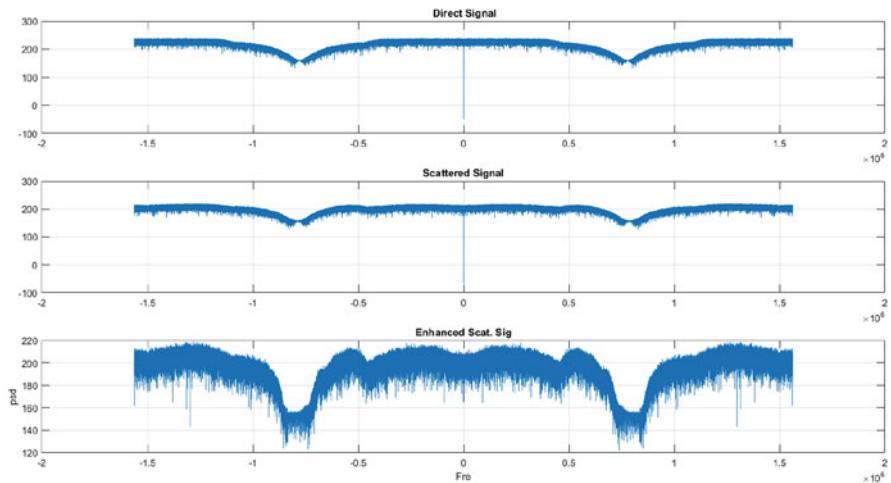


Fig. 27.11 Adaptive spectral radar signal analysis

References

- [Mahafza2008] Mahafza, B. R., Radar Signal Analysis and Processing Using MATLAB, Chapman and Hall/CRC, 2008
- [Budge2015] Budge, M.C., German, S.R., Basic Radar Analysis, Artech House, 2015
- [Kang2008] Kang, Euyng W., Radar System Analysis, Design and Simulation, Artech House, August, 2008
- [Kolawole2002] Kolawole, M. O., Radar systems, peak detection and tracking, Elsevier Ltd., 2002
- [Skolnik2008] Skolnik, M., Radar Handbook, Third Edition, Mc Graw Hill, 2008
- [George2007] George J. M., Multi-Dimensional Data Transmission, Artech House, 2007
- [Li05] Li, H.-J., Kiang, Y.-W., Moving Target Indicator Radar and Inverse Scattering; The Electrical Engineering Handbook, Elsevier, 2005
- [zamani2017] Zamani, M., Abbas Sheikhi, A., “Mixed Signal-Based GLR Detector for FM Passive Bistatic Radar Target Detection,” Progress In Electromagnetics Research M, Vol. 55, 37–49, 2017. <https://doi.org/10.2528/PIERM16122501>, <http://www.jpier.org/PIERM/pier.php?paper=16122501>
- [Theodoridis14] Theodoridis, S., R Chellappa, R., Communication and Radar Signal Processing, Elsevier, 2014
- [Pouliarikas18] Pouliarikas, A., Dorf, R. C., Advanced Signal Processing Handbook, CRC Press, 2018

Chapter 28

Visual Story Telling



Overview

Visualization is a form of communication. Visual Story Telling (VST) is a visual perception of the data. An informed decision about the situation can be made through an effective communication with the data. Understanding the context, selecting the right tool to display, right selection of data visualization by removing redundancy, orientation, and focus are important for data visualization. Visual forms such as charts, graphs, color, maps, bars, statistics are typically applied to describe the data. Such visual representation can be enhanced by using graphics, music, voice, different forms of audio, video, text and sound. Human processes information using the perceptual ability of vision, even before the complex cognitive processes of the human mind come into play. When we talk about grasping the information, the human visual perception is stronger than the information reception only. Visual analytics (VA) can be seen as a bridge between cognition and perception. This means the VA helps in understanding complex ideas, their relationships, compositions, distributions, comparisons, finds information that otherwise might remain hidden in the data and present them using different graphics or visual forms. The **Visual Story Telling** applies concepts on classification, clustering, dimensionality reduction, regression; information exploration and statistical computations. In this chapter, VA, VST, and data analysis are briefly introduced. These fields are interrelated.

28.1 Introduction

Visualization is the graphic representation of data to reflect a complex information. Right tools and techniques are necessary to reveal insight and extract desired information generate. A fundamental question in data visualization is how to communicate with the data that leads one to make the decision. This has several

steps: Objective, preparation, analysis and presentation. Being able to visualize data and tell stories with data can help to make an informed decision.

Visualizations lead to discover interesting patterns into actionable insights quickly and intuitively, also to communicate these insights to assist in making decisions.

Data comes with varieties such as structured, unstructured, static, or video stream. Visualization transforms the information in the data into visual representation using visual forms, images and graphics. The VST is to present and interpret the world from collected data using visual elements and symbols so that the sensible and meaningful stories can be revealed and predicted. The visual transformation of the data is based on data, information, known theories, knowledge, belief, values, judgement, knowledge discovery applying predictive analytics, data mining by interactive visualization.

The one objective of the visual abstraction, insight of the data using visual forms and patterns, transformation to predictive analytics in VST is to assist in situation awareness, prediction and to make informed decision.

The VST has two major components:

- Visualization
- Visual Data Mining

Common four steps in both components are data collection, visualization, cognition and perception, as well as prediction. Graphs, charts, tables or networks are some typical visual data exploration forms. The visual information is further processed for cognition such as learning from the data and recognize patterns. In the cognition step, features are extracted from the data through data transformation. These features are then analyzed for pattern extraction in the perception process. The patterns are used for classification and knowledge extraction. Thus the knowledge is used to predict an occurrence or in an emergency operation centre.

The VDM is a young and emerging discipline that combines knowledge and techniques from a number of areas. One main goal of VDM is to devise visualizations of large data into interpretations. The VST processing steps are:

- Visualization: This transforms data into visual representations. This conveys the data abstraction as an initial stage of communication. This has following steps:
 - Explanatory: This gives an insight of the data for visualization. This conveys messages to viewers. This encodes data into visualizations that conveys messages inherent in the data.
 - Exploratory: Visual interaction is the data exploration for feature extraction. This is the information transformation and extraction through feature transformation. This predicts the new information.
 - Confirmatory: In this stage, the hypothesis of new information is evaluated for any particular phenomenon.

- Prediction: This is the knowledge extraction step for making decision. This is done by capturing abstract data, encoding the data into visualization interactions, transforming the visual representation into features, finding the patterns and feature space for classification and discovery of any new knowledge. This can be explained applying visual data mining (VDM).
- When happened?
- Where that happened?
- What is happening now?
- What is going to happen?
- Take action based on the above fours

28.1.1 Common Visualization Approaches

Common visualization approaches are line chart, bar chart, column chart, pie chart, area chart, pivot table, scatter chart, bubble chart, tree map, polar chart, area map, scatter map. 2D area can be visualized by cartogram, choropleth, dot distribution map; temporal visualization can be visualized using connected scatter plot, polar area diagram, time series, and multi-dimensional data elements can be visualized by pie chart, histogram, hierarchical data set can be seen using dendrogram, ring chart, tree diagram, network related datasets can be seen using alluvial diagram, node-link diagram, matrix. Some conventional statistical computations are often powerful enough to provide an insight of data. A popular dominantly visualization software called D3 often makes the data immediate and interactive.

28.2 Analytics and Visualization

A science of visual representations using visual representations as needed to address the diverse data types that are relevant to the problems, or the subject matters, has been developed to support scientific applications, is the visual analytics. The goal of visualization in computing is to gain insight by using our visual machinery, a method of seeing the unseen, fosters profound and unexpected insights, maximize human understanding, and communication, gain understanding and insight into the data, a deeper level of understanding and introduce new insight, to gain insight into an information space, to assist in perception, reduce cognition load.

28.2.1 *Exploratory Data Analysis*

28.2.1.1 **Exploratory Data Analysis**

The Exploratory Data Analysis (EDA) is a visual statistical approach for gaining insight of the data. This is necessary to gain intuition, to build meaningful features and select for right model, to generate hypothesis, and insight of the data.

Understanding data using the EDA for the visualization, and then predict them for a future occurrence of an event is the visual analytics. After data analysis, visualization with basic statistical computation, and visualization methods, processed, the analytical methods are applied. The analytical methods can be descriptive, predictive, perspective and optimization. The visual data analytics can be descriptive, predictive, prescriptive. The descriptive data systems and their preparation including databases, and warehousing, data cleaning and engineering, and data monitoring, reporting, visualization. The second aspect involves data analytics and includes and predictive analysis and for prescriptive analysis and optimization. Descriptive analytics learns from past behaviors, and describes the influence of future outcomes applying the learned behaviors. The descriptive analytics is predominantly used in statistics. Descriptive analysis describes the statistics of the data, and summarizes the raw data and makes them human interpret able to the human being. This is the analytics that describes the past. The past refers to any point of time that an event has occurred, whether it is one minute, or one year ago.

Descriptive Analytics is used to understand the aggregation about on going status, to summarize and describe different aspects of the situation based on the available resources Predictive analytics uses statistical models and forecasts techniques to understand the future and answer, e.g., “What could happen?”. Predictive analytics is the process of discovering interesting and meaningful patterns in data. The methods are taken from pattern recognition, machine learning, artificial intelligence, and data mining.

Predictive analytics has its roots in the ability to “Predict” what might happen. These analytics are about understanding the future. This provides actionable insights based on data. Predictive analytics gives the likelihood of a future outcome. No statistical algorithm can “predict” the future with 100% certainty, hence the use of predictive analytics to apply statistics to forecast what might happen in the future. The predictive analytics is based on probabilities. A common application of predictive analytics is to produce a credit score. These scores are used by financial services to determine the probability of customers making future credit payments on time.

Prescriptive analytics prescribes possible outcomes. This uses optimization and simulation algorithms for possible outcomes and answer, i.e., what should we do. Prescriptive analytics attempt to quantify the effect of future decisions in order to advise on possible outcomes before the decisions are actually made. Prescriptive analytics predicts not only what will happen, but also why it will happen providing recommendations regarding actions that will take advantage of

the predictions. These analytics go beyond descriptive and predictive analytics by recommending one or more possible courses of action. Essentially they predict multiple futures and allow companies to assess a number of possible outcomes based upon their actions. Prescriptive analytics use a combination of techniques and tools such as business rules, algorithms, machine learning and computational modelling procedures. These techniques are applied against input from many different data sets including historical and transactional data, real-time data feeds, and big data.

28.2.2 *Visual Data Mining*

Data mining is an analytical process based on mathematical algorithms and statistics. Data exploration, patterns discovery, and relationships are preliminary data mining themes. can be as exciting as they are challenging; analysts are rewarded with a progressively evolving list of questions to be answered as the data reveal additional insights and relationships.

It is better to develop the ability to work with and through data in an effort to reveal possible underlying trends and patterns. This is a craft in some ways. In my experience, there is a degree of creativity in the type of pattern recognition that is associated with a good analyst: the ability to see the meaning hidden among the general disorder in the information. Data mining truly can be described as a discovery process. It is a way to illuminate the order that often is hidden to all but the skilled eye.

28.3 Communication and Visualization

Visualization is a communication mode that humans use for interactions. Often symbolic pictures on rules and regulations in business or in streets are more interactive, and instantaneous than some blocks of texts. Graphical instructions are easy to understand and to act quickly, immediate, powerful, memorable. Thus visualization is less intense to human cognition than listing or writing or reading text or material.

Visualization spots patterns—distribution, clusters, anomalies, correlation; stimulates and visually evaluates hypotheses. Visualization can be formed when the input data is in textual, binary or different network type, tabular, software, volumetric, mathematical such as vector field, tensor field, geo-information, bio-information. How does visualization saves time: overview of the time and attributes together; omission of seeing time and without seeing the time—video and animation; external memorization for more useful brain work—information display using events, timeline and occurrences; how much and how to display information, generate hypothesis for intuition, experience, and knowledge.

Exploratory data analysis is normally first step in data analysis. In exploratory data analysis, the collected data is visualized using their distribution, the information about the variables at a preliminary level and to see patterns, that leads to idea, and vice versa. Patterns lead to question, and hypothesis testing generates when idea leads to visualization. Statistical computations and visual graphics such as mean, median, box plot, scatter plot, windowing technique give the preliminary information about the data. The next steps are feature selection and feature analysis for building a right model. The EDA reveals the data information namely value types, distribution, relationships, ideas about prediction problem. A very beginning task is framing the process and problem of the analytics by identifying the users. This includes idea about the desired outcome, available data sources, influence and impact factors, decision making capability with respect to time, available resources, risk, problem definition and information about the data.

Example: Adapted example of visualization in COVID-19 cases in Canada done by <https://covid-19-canada.uwo.ca/en/data.html> using matplotlib (Figs. 28.1, 28.2, 28.3) is presented here. In these figures, we see an overview of the pandemic situations with respect to mortality and recovery rate of COVID-19 situation in Canada based on the data published from Government of Canada. Cumulative and daily increased cases about spreading the coronavirus and the slowing down process between 4/7/2020 and 4/17/2020 and cumulative recovered cases remains unchanged between 4/7/2020 and 4/17/2020 and reported missing dataset is also shown. The decrease of unchanged cumulative recovery rate during this period is shown in the figure.

The first step is to get an insight of a situation is to understand the data and the context.

“Visual form” refers to the geometry, colour, texture, brightness, contrast and some visual attributes that characterise and influence the visual perception of an object. Thus, the source space is the space of 2D and 3D shapes and the attributes of their visual representation. The generalisations of patterns discovered in data are described using the subject domain concepts associated with the domain functions constructed from the domain vocabulary.

28.4 Background Information

Automatic analysis techniques such as statistics and data mining developed independently from visualization and interaction techniques. The visual analytics combines automatic and visual analysis methods to gain knowledge from data and transform them into human machine interactive mode. The exploratory data analysis based on statistic research is discussed by John W. Tukey in his book “Exploratory data analysis”. The research areas are expanding rapidly in different parts of the world; a greater portion of a research community is involved in information visualization

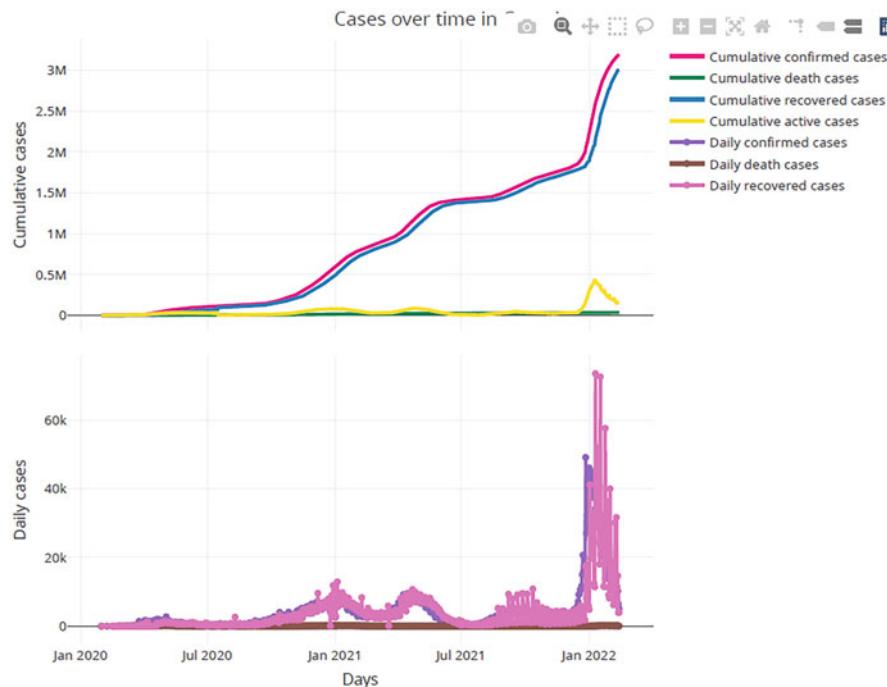


Fig. 28.1 Image 28.1.: Covid-19 cases over time in Canada (<https://covid-19-canada.uwo>)

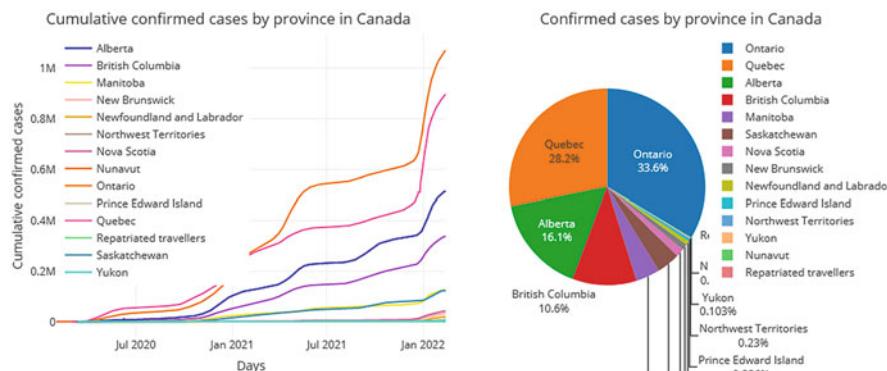


Fig. 28.2 Image 28.2.: Cumulative confirmed cases (<https://covid-19-canada.uwo>)

(Keim et al. 2009). Visualization is the graphical representation of data on the basis of Visual analytics. Data collections, modelling, and storing information need analytical reasoning supported by interactive visual interfaces. Complexities of massive data a rapidly growing rapidly to analyse them properly using the right tools and techniques. Day by day, technologies generate a variety of complex data

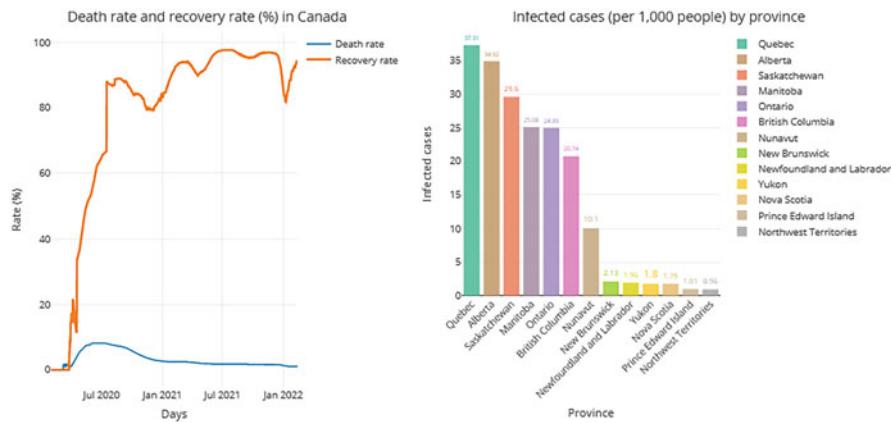


Fig. 28.3 Image 28.3.: Visualization of Covid-19 cases using exploratory data analysis (<https://covid-19-canada.uwo>)

sets from different data sources in increasing volumes, and with more velocity. Mining of these large data sets requires efficient data organization, visualization and representation.

The VST is an interdisciplinary area of statistics, data visualization, data mining, artificial intelligence, machine learning, stochastic process, data fusion, cognition science. The research started from the need of exploratory data analysis. This led to visual data exploration and visual data mining. The extensions of this are since then continuing in information visualization, newly invented techniques in data mining to explore the data in a more appealing and visually interactive manner. The visual analytics is the one of the most dominant areas in assisting data learning and next steps. Data visualization gives an insight i.e. shows the unseen facts, reveals hidden and significant events, maximizes the deeper level of understanding into data, and promotes a better communication through new insights with them into information space. Thus, the visualization and visual explanations amplify cognition through graphics and predictions. At the same time, the reasoning process to understand situations, enable actions, and enhance actions planning through taking control over situations are also intensified.

The objective of training a neural network with data, that is, to determine its weights on the basis of the available data set, is not to find an exact representation of the training data, but to build a model that can be generalised or that allows us to obtain valid classifications and predictions when fed with new data.

This chapter has provided an essence of the topic, the details with case studies and examples are ongoing and will be available in our future endeavour.

28.5 Exercises

Exercise 1 Use a traffic and traffic accident data set: (a) Create a chart that shows the variability in website traffic and traffic accident for each day of the week;

- (a) Use the same data set for the temporal data visualization.
- (b) Use the same data set, predict the traffic jam and accident at several places.

References

- [**Mazza2009**] Mazza, R, Introduction to Information Visualization, Publisher Springer London, February, 2009
- [**McCue2015**] McCue, C., Data Mining and Predictive Analysis, Elsevier, 2015
- [**Miles2020**] Miles, J.; Turner, J., Jacques, R., Williams, J., Mason, S., Using machine-learning risk prediction models to triage the acuity of undifferentiated patients entering the emergency care system: a systematic review, Diagnostic and Prognostic Research, Springer Nature, volume 4, Article number: 16, 2020
- [**Keim2009**] Keim, D. A., Mansmann, F., Stoffel, A., Ziegler, H., Visual Analytics, Encyclopedia, Springer, 2009
- [**Tukey77**] J. W. Tukey, J. W. Exploratory Data Analysis. Addison-Wesley, Reading MA, 1977
- [**UWO22**] Data Visualization, Western Science, University of Western Ontario, <https://covid-19-canada.uwo.ca/en/data.html>, 2022

Chapter 29

Digital Processes and Multimedia



Overview

There is no precise definition of multimedia. Often, signal processes do represent multimedia. These representations may be 2-dimensional, 3-dimensional, or even more complex. For humans they are addressed to the eye, to the ear, or to a combination of both. More esoteric interfaces may address touch or even directly the brain. A pretty comprehensive list of signal processing topics related to multimedia systems includes: concepts and principles of multimedia systems; speech analysis and recognition; audio/image/video compression; scene video analysis and understanding; multimedia applications in human computer interaction, multimedia communication and multimedia security. A simple example of multimedia is a web page with two-dimensional images and sound.

29.1 Images

Images provide examples that present many problems which are typical when considering multimedia. The objective of image understanding is a symbolic representation of the contents of an image. Applications of image understanding are computer vision and robotics. In this chapter, we will provide examples and applications in image enhancement, restoration and image information retrieval. To achieve the stated goal, we mainly use techniques from signal processing and machine learning.

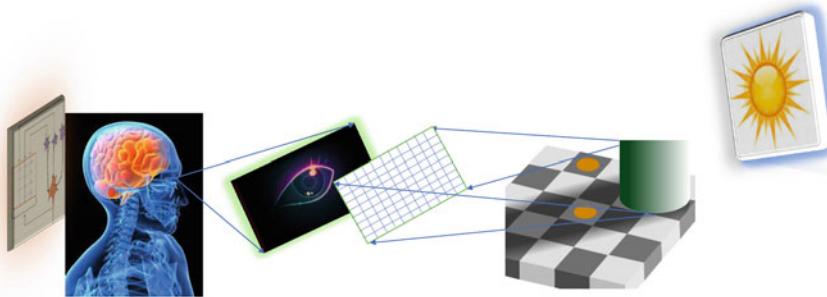


Fig. 29.1 Human image perception

29.1.1 *Digital Image Processing*

A digital image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels. Pixel values typically represent gray levels or colors as depicted in the Fig. 29.1.

The computer receives an image in the form of signals that are representations of pixels, e.g., Fig. 29.2. Pixel values may be represented as 8-bit numbers (for gray scale image representation) or as tuple of three values, red-R, green-G, and blue-B (RGB) per pixel each representing an “additive” (as opposed to “subtractive”) color.

A image can be seen as a collection of elements organized in a MATRIX form; clearly indicated in Fig. 29.2. Each Matrix element represent the smallest part of a picture element, that is a pixel. Images are thus characterized by the number of pixels; the larger the number of pixels the better the quality of an image. However, the larger the number of pixels the larger the storage space required to store the image. In order to better comprehend why those tuples are used to represent a color image we are providing a historical perspective of our present understanding of visual perception. From the studies conducted in early 1930s, clearly only a small sliver of wavelengths is visible to human eye as depicted in Fig. 29.3. Humans have been able to convert analog image into digital form as described in earlier chapters. Fundamental concept of pixel has been previously described. What information is contained in pixels? If the information is represented as a 4-bit code than there could be only $2^4 = 16$ different representations. Typically 8 bits are used to represent each pixel. This defines a $2^8 = 256$ levels of representation. What does this level represent? It can represent:

- **Gray Scale**—see Fig. 29.4.
- **Color Scale**—There are several ways to represent color that gives raise to various standards that are going to be discussed next.

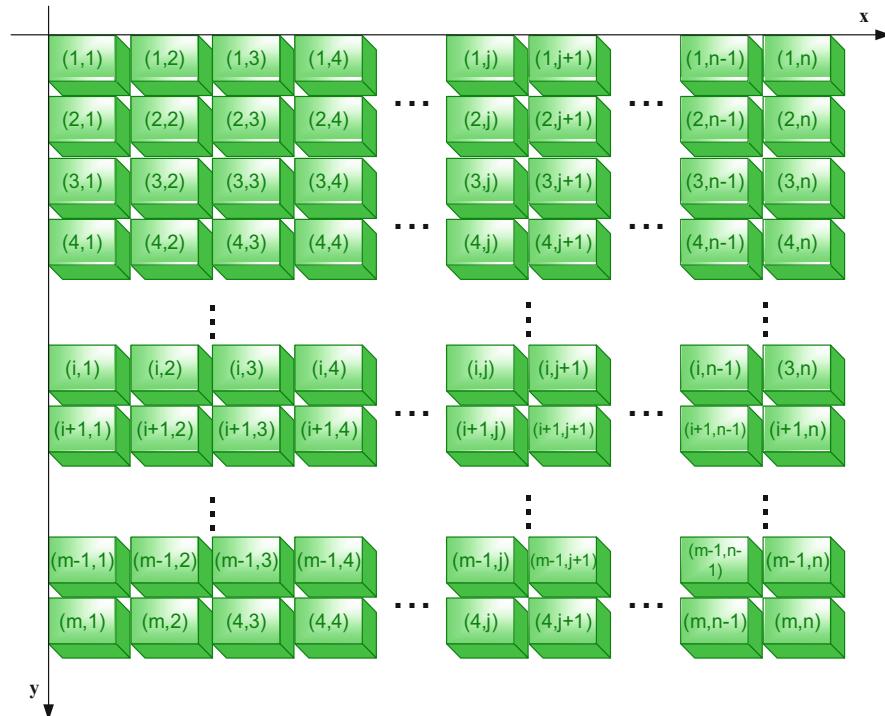


Fig. 29.2 Digital image representation

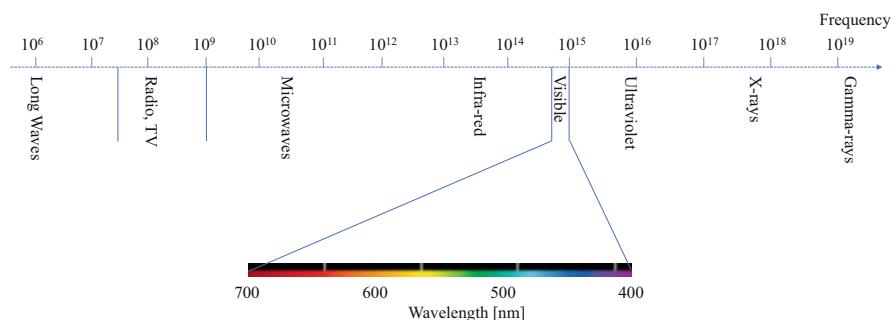


Fig. 29.3 Visible wavelengths

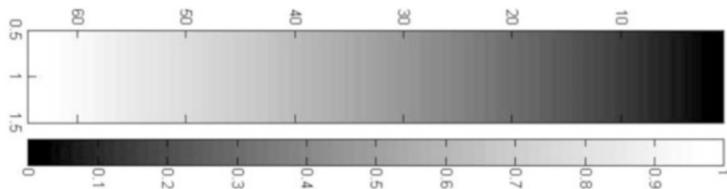


Fig. 29.4 Gray Scale representation (MATLAB)

Monochromatic Scale

The idea behind Gray scale image representation is to represent the image in a monochromatic scale. Hence, Gray scale image may be thought of as being defined by a single quantity/measurement calling it “quality” of a single “color”.

Color Scale

From theory of light and the theories of how humans perceive color information there are a number of standards that have been developed to represent color. They all address the Chromaticity—defined as the quality of a color as determined by its “purity” and dominant wavelength.

A color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers (e.g. three numbers in RGB or four as in CMYK representation). This representation was inspired from human anatomy (eye) and the way the colors are perceived. Human perception of colors is described by “color space”. The human eye has receptors for short (S), middle (M), and long (L) wavelengths, also known as blue, green, and red receptors. That means that one, in principle, needs three parameters to describe a color sensation.

In the study of the perception of color, one of the first mathematically defined color spaces was the CIE XYZ color space (also known as CIE 1931 color space), created by the International Commission on Illumination (CIE) in 1931. It is common practice to define pure colors in terms of the wavelengths of light as shown in Fig. 29.3. This works well for pure spectral colors but it is found that many different combinations of light wavelengths can produce the same perception of color. This progression from left to right is from long wavelength to short wavelength, and from high frequency to low frequency light. The wavelengths are commonly expressed in nanometers ($1 \text{ nm} = 10^{-9} \text{ m}$). The visible spectrum is roughly from 400 nm (violet-end) to 700 nm (red-end).

A specific method for associating three numbers (or tristimulus values) with each color is called a color space: the CIE XYZ color space is one of now many such spaces. However, the CIE XYZ color space is special, because it is based on direct measurements of the human eye, and serves as the basis from which many other color spaces are defined.

Common image formats include:

- 1 sample per point (Black—White or Grayscale)
- 3 samples per point (Red, Green, and Blue)
- 4 samples per point (Red, Green, Blue, and “Alpha”, a.k.a. Opacity), etc.

An image can be regarded as a function:

$$\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

Where $f(x, y, z)$, gives the “intensity” at the position of (x, y, z) point that is projected in 2D image (x^P, y^P) . Since an image can be considered as mapping from three dimensions into 2 dimensions, function that gives that value we termed it with f , and it can be represented as Grayscale or RGB, or CMYK, or \dots whatever representation we chose.

Example of Grey-scale image representation:

$$\mathbf{f} : [x, y, z] \rightarrow [x^P, y^P][0, 255]$$

and RGB image representation, from original three dimensional image f .

$$\mathbf{f} : [x, y, z] \rightarrow [x^P, y^P] \quad \mathbb{R}[0, 255], \quad \mathbb{G}[0, 255], \quad \mathbb{B}[0, 255]$$

Since the human eye has three types of color sensors that respond to different ranges of wavelengths, a full plot of all visible colors is a three-dimensional figure. However, the concept of color can be divided into two parts: brightness and chromaticity. For example, the color white is a bright color, while the color grey is considered to be a less bright version of that same white. In other words, the chromaticity of white and grey are the same while their brightness differs.

The CIE XYZ color space was deliberately designed so that the Y parameter was a measure of the brightness or luminance of a color. The chromaticity of a color was then specified by the two derived parameters x and y which are functions of all three tristimulus values X, Y, and Z.

A color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components (e.g. RGB and CMYK are color models). Any color which can be produced by the primary colors blue, green, and red can be written:

$$\mathbf{C} = \mathbf{R} \vec{R} + \mathbf{G} \vec{G} + \mathbf{B} \vec{B}$$

where \vec{R} , \vec{G} , and \vec{B} be considered to be “unit values” for red, green, and blue, and R,G,B are the magnitudes or relative intensities of those primaries and are called “tristimulus values”. Note that the “unit values” associated with \vec{R} , \vec{G} , and \vec{B} are of different size in physical power units (watts) because the sensitivity of the eye will be different for the different primary colors. Note that if a different set of primary colors is chosen, the unit values necessary to produce white in mixture would have to be re-established. In modern color measurement the CIE tristimulus values are probably the most important.

An RGB color space is any additive color space based on the RGB color model. RGB is a convenient color model for computer graphics because the human visual

system works in a way that is similar—though not quite identical to an RGB color space. The most commonly used RGB color spaces are sRGB and Adobe RGB (which has a significantly larger gamut). Adobe has recently developed another color space called Adobe Wide Gamut RGB, which is even larger, in detriment to gamut density. As of 2007, sRGB is by far the most commonly used RGB color space, particularly in consumer grade digital cameras, because it is considered adequate for most consumer applications, and its design simplifies previewing on the typical computer display. RGB spaces are generally specified by defining three primary colors and a white point. In the table below the three primary colors and white points for various RGB spaces are given. The primary colors are specified in terms of their CIE 1931 color space chromaticity coordinates (x,y) (Fig. 29.5).

However, there are many other color spaces; like for example It is possible to achieve a large range of colors seen by combining cyan, magenta, and yellow transparent dyes/inks on a white substrate. These are the subtractive primary colors. Often a fourth black is added to improve reproduction of some dark colors. This is called “CMY” or “CMYK” color space. The cyan ink will reflect all but the red light, the yellow ink will reflect all but the blue light and the magenta ink will reflect all but the green light. This is because cyan light is an equal mixture of green and blue, yellow is an equal mixture of red and green, and magenta light is an equal mixture of red and blue.

Computer monitors are based on RGB color scale. The typical color computer monitor can be thought of as being presented in Fig. 29.6:

Standard wide screen computer Monitor in the year 2010 was:

- 1920×1080 pixels
- $1,296,000 \approx 1.3M$ pixels
- 32-bit RGB Representation
- Total: $1,296,000 \times 32 \times 3 = 124,416,000 \approx 124M\text{bits}$ or $24,883,200 \approx 25M\text{Bytes}$

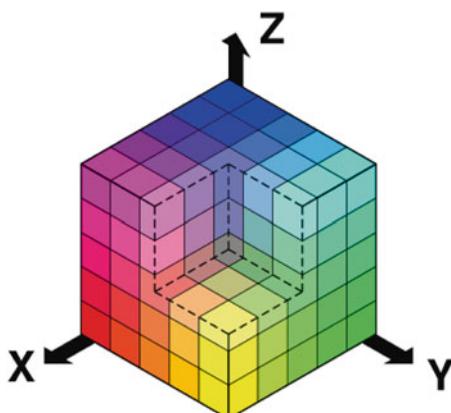


Fig. 29.5 RGB color space

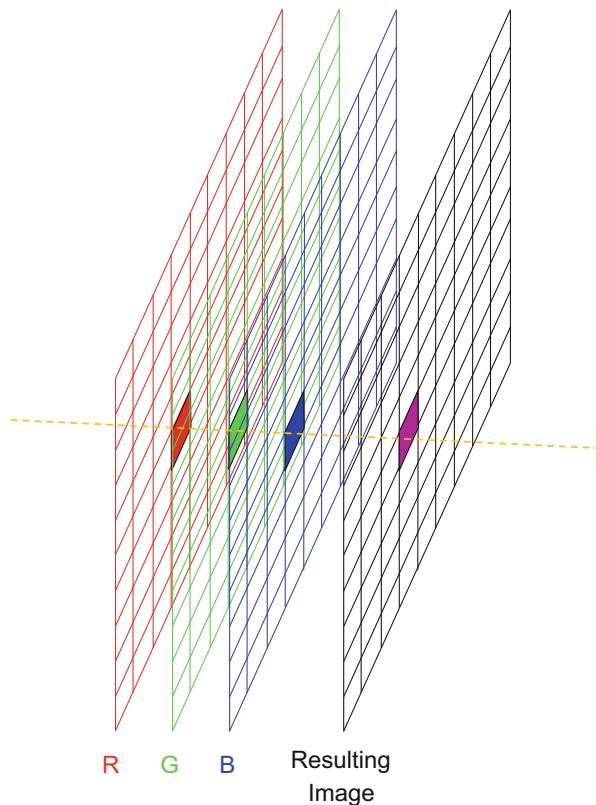


Fig. 29.6 RGB Color representation of a computer image

Colored image is a function of typically three channels (R, G, B) and it can be written as:

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \mathbf{r}(\mathbf{x}, \mathbf{y}) \\ \mathbf{g}(\mathbf{x}, \mathbf{y}) \\ \mathbf{b}(\mathbf{x}, \mathbf{y}) \end{bmatrix}$$

The digital image thus can be considered as being composed of three one colored channels as depicted in the Fig. 29.7:

The same images were displayed individually with the color that it composes it (e.g., R, G and B) (Fig. 29.8).

The same images were displayed individually with the color that it composes it (e.g., R, G and B).



Fig. 29.7 Individual RGB colors of an image



Fig. 29.8 Individually depicted RGB colors of an image

General focus of Digital Image Processing is in two major areas:

- (1) Improving on image representation for human consumption, and
- (2) Processing the image data for storage, transmission, and representation for autonomous machine perception.

One recognizes three levels of image processing:

- (1) Low Level

- (a) Input: **Image**
- (b) Output: **Image**
- (c) Examples:
 - **Noise Removal**
 - **Image Sharpening**
 - **Contrast Enhancement**
 - ...

- (2) Mid Level,

- (a) Input: **Image**
- (b) Output: **Image Attributes**
- (c) Examples:
 - **Object Recognition**
 - **Image Segmentation**
 - ...

and

- (3) High Level

- (a) Input: **Image Attributes**
- (b) Output: **Image Understanding**
- (c) Examples:
 - **Scene Understanding**
 - **Autonomous Navigation**
 - ...

29.1.2 *Images as Matrices*

An image matrix ($N \times M$):

$$\mathbf{A} = \begin{bmatrix} A(0, 0) & \cdots & A(0, N-1) \\ \vdots & \ddots & \vdots \\ A(N-1, 0) & \cdots & A(M-1, N-1) \end{bmatrix}$$

with elements $A(i, j) \in \{0, 1, \dots, 255\}$ and each element $A(i, j)$ has a dual interpretation:

- “Matrix”—The matrix element (i, j) with value of $A(i, j)$
- “Image”—The pixel (i,j) with value $A(i, j)$

The representation of an $M \times N$ numerical array (e.g., matrix) as an:

$$\mathbf{f}(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \cdots & f(M-1, N-1) \end{bmatrix}$$

However, the representation of a $M \times N$ numerical array is typically written as:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

The number b of bits is required to store a $M \times N$ digitized image:

$$b = M \times N \times k$$

with $L = 2^k$ representing number of (gray scale, or tri-stimulus values as in RGB) levels.

29.1.3 Gray Scale Images

As indicated earlier, the gray scale image is considered a representation of only one color (or lack of it). Below are three images represented in gray scale (Fig. 29.9).

Matlab script that was used to generate images is presented below:

MATLAB Script:

```
M &= 256;
N &= 256;
for i=1:N
    for j=1:M
        A(i,j)= (j-1);
    end
end
subplot(1,3,1);
imshow(uint8 (A));
```

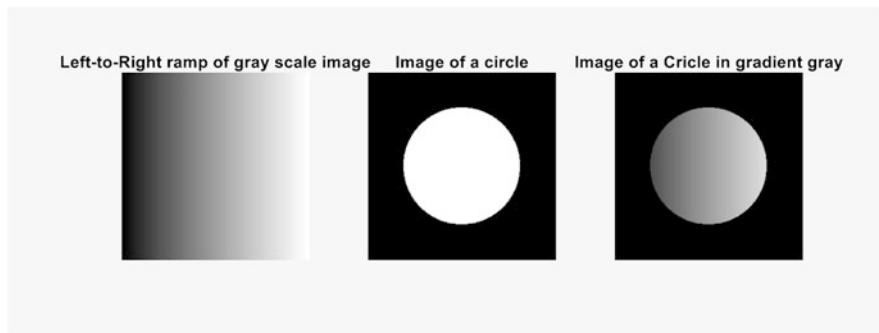


Fig. 29.9 Gray scale image representation

```
title("Left-to-Right ramp of gray scale image");
colormap gray;
axis('image');
for i = 1 : N
    for j = 1 : M
        dist = uint8(((i-N/2)^2+(j-M/2)^2)^0.5));
        if (dist < 80)
            B(i,j) = N;
        else
            B(i,j) = 0;
        end
    end
end
subplot(1,3,2);
imshow(B);
title("Image of a circle");
colormap(gray);
axis('image');
C = A;
for i = 1 : N
    for j = 1 : M
        C(i, j) = A(i, j).*B(i, j)/N;
    end
end
subplot(1,3,3);
image(C);
colormap(gray);
axis('image');
axis off;
RGB8 = uint8(C);
imshow(RGB8);
```

```
colormap(gray);
title("Image of a Circle in gradient gray");
axis('image');
```

29.2 Spatial Filtering

The name “spatial filtering” is borrowed from frequency domain processing. This processing refers to: Passing, Modifying, or Rejecting specified frequency components of an image. A filter that passes low frequencies is called a low-pass filter. This process of applying filters in the image we call “spatial filtering”. Application of spatial filter in the image will replace its original values by a function based on the pixel value and its neighbors. If the operation performed is linear then filter is called a linear spatial filter. Otherwise the filter is a nonlinear spatial filter.

29.2.1 *Linear Filtering of Images*

We are going to present a method of linear filtering applied to images. 2D Linear filtering is the operation that it is applied to the image $x(k, l)$ to obtain the filtered image $y(k, l)$ defined by:

$$y(k, l) = x(k, l)h(k, l) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} x(k-m, l-n)h(m, n)$$

The sequence $h(k, l)$ dependent on indexes that defines a filter and is called the Point Spread Function (PSF). As was the case with 1D (e.g., sequences of time) filtering, the convolution operation is linear and space-invariant. The $h(k, l)$ is the equivalent of the impulse response of the one dimensional filter. If x , h and y are related by the following:

$$y(k, l) = x(k, l) * h(k, l)$$

Then the X , H and Y ; the discrete frequency representations of the time domain signals x , h and y , are related by the following expression in case of linear and time invariant system:

$$\mathbf{Y}(\mu, v) = \mathbf{X}(\mu, v)\mathbf{H}(\mu, v)$$

where X , H and Y are 2D Fourier Representations of x , h and y respectively.

29.2.2 Separable Filters

A filter is said to be separable if its PSF has the following property:

$$\mathbf{h} = \mathbf{h}_x \mathbf{h}_y^T$$

In case of finite PSF represented by matrix h of elements $h(k, l)$ and if h_x and h_y are the vectors with respective components $h_x(k)$ and $h_y(l)$ then the relation above is equivalent to:

$$h(k, l) = h_x(k)h_y(l)$$

Hence the filtering operation can be applied independently of the direction as indicated below:

$$\begin{aligned} y(k, l) &= x(k, l) * h(k, l) \\ &= \sum_{m=K_1}^{K_2} \sum_{n=L_1}^{L_2} x(k-m, l-n)h(m, n) \\ &= \sum_{m=K_1}^{K_2} \sum_{n=L_1}^{L_2} x(k-m, l-n)h(m)h(n) \\ &= \sum_{m=K_1}^{K_2} h(m) \left(\sum_{n=L_1}^{L_2} x(k-m, l-n)h(n) \right) \\ &= \sum_{n=L_1}^{L_2} h(n) \left(\sum_{m=K_1}^{K_2} x(k-m, l-n)h(m) \right) \end{aligned}$$

29.2.2.1 Gaussian Filter

Gaussian filter is a separable filter as shown next:

$$\begin{aligned} h(k, l) &= \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \frac{k^2+l^2}{\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{k^2}{\sigma^2}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{l^2}{\sigma^2}} \\ &= h(k)h(l) \end{aligned}$$

Gaussian Kernel smoothing filter (Fig. 29.10):

Similarly to time domain windowing functions, the filtering operation in 2D, or other higher dimensional spaces, are based on windowing of corresponding dimensions.

Sum-of-products operation between an image x and a filter kernel, h defines the linear filtering. The kernel is an array whose size defines the neighborhood of operation, and whose coefficients determine the nature of the filter. Other terms used to refer to a spatial filter kernel are: Mask, Template, and Window (Fig. 29.11).

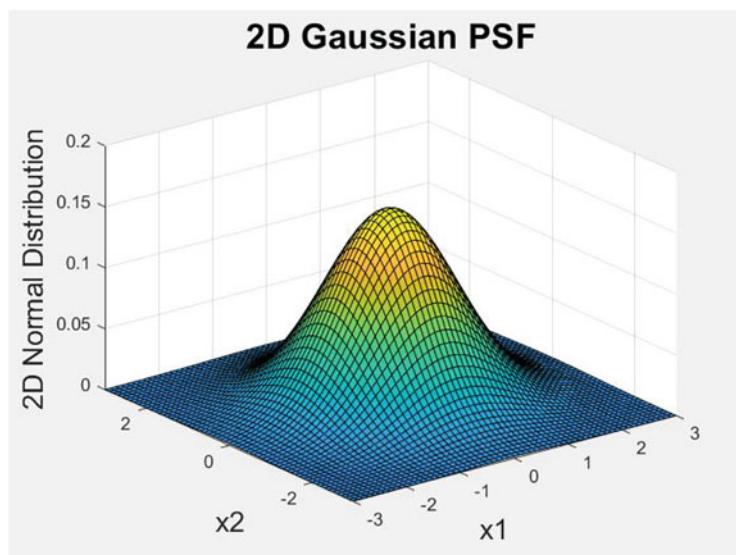


Fig. 29.10 Gaussian Smoothing Kernel

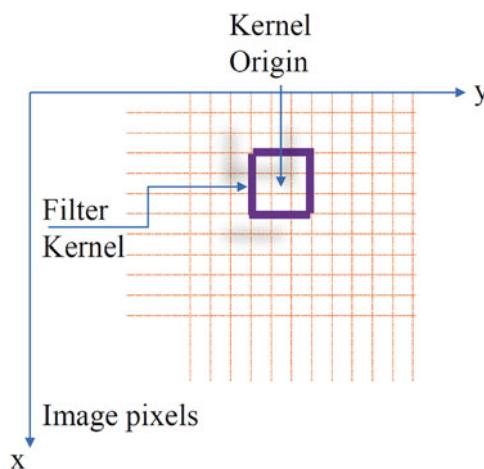


Fig. 29.11 Discretization of an Image and Filter Kernel

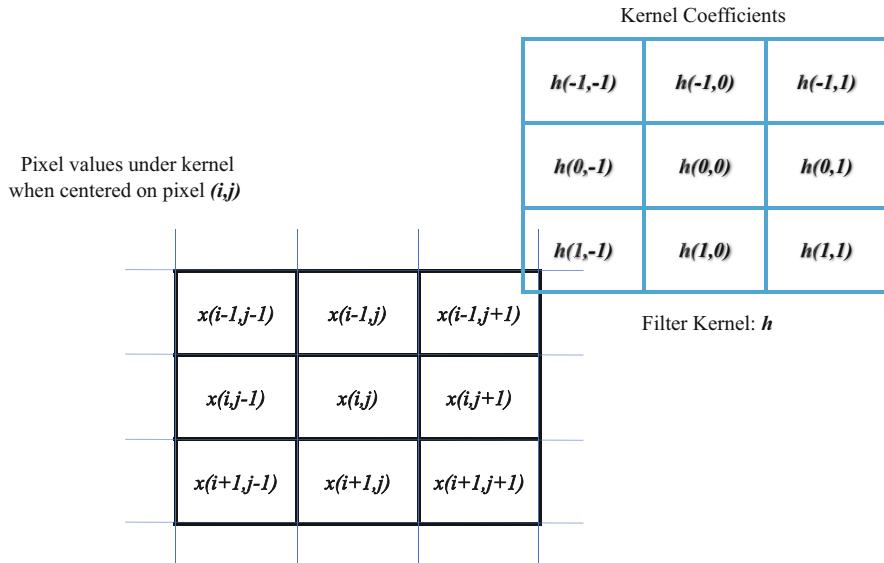


Fig. 29.12 Filtering operation

The filtering operation is defined with the example of filter kernel of size (9×9) as depicted in the Fig. 29.12:

29.2.3 Mechanics of Linear Spatial Filtering Operation

In the previous figure, an application of the spatial filtering using a 3×3 kernel h is exemplified. At any point (i,k) in the image, the response $y(i, k)$ of the filter is applied as the sum of products of the kernel coefficients and the image pixels covered by the kernel:

$$\begin{aligned}
 y(i, k) = & h(-1, -1)x(i - 1, j - 1) + h(-1, 0)x(i - 1, j) + \dots \\
 & + h(0, 0)x(i, j) + \dots \\
 & + h(1, 1)x(i + 1, j + 1)
 \end{aligned}$$

As coordinates i and j are necessarily varied, the center of the kernel moves from one pixel to the other, generating filtered image y . One could clearly observe that the center coefficient of the kernel, $h(0, 0)$ aligns with the pixels at the location (i, j) . For a kernel of size $m \times n$, we assume that $m = 2a + 1$ and $n = 2b + 1$, where a and b are nonnegative integers. This means that our focus is on kernels of odd size in both coordinate directions. Linear spatial filtering of an image of size $M \times N$ with a kernel of size $m \times n$ is given by expression:

$$y(i, j) = \sum_{s=-a}^a \sum_{t=-b}^b h(s, t)x(i + s, j + t)$$

Where i and j are varied so that the center of the kernel visits every pixel in x once.

In theory, the operation that it is applied to the image $x(k, l)$ to obtain the image $y(k, l)$ is defined by:

$$y(k, l) = x(k, l) * h(k, l) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} x(k - m, l - n)h(m, n)$$

where with $h(k, l)$ is Point Spread Function (PSF) that is dependent on indexes that define a filter. As was the case with time dependent (e.g., 1D) sequences filtering the convolution operation is linear and space-invariant, and $h(k, l)$ is the equivalent of the impulse response of the one-dimensional (e.g., 1D) case.

Some important concepts from the Time Domain vs. Frequency Domain that will help in mastering the presented material is in order. Fourier Transform links the concepts of spatial with frequency domain. We return to spatial domain via Inverse Fourier transform. The fundamental properties relating the spatial and frequency domains are:

- (1) Convolution, which is the basis for filtering in the spatial domain is equivalent to multiplication in the frequency domain and vice versa.
- (2) An impulse of strength \mathbf{A} in spatial domain is a constant value \mathbf{A} in the frequency domain and vice versa.

The appearance of the image depends on the frequency's of its sinusoidal components, and the change in the frequencies of those components will change the appearance of the image. This is a powerful concept—it makes it possible to associate certain frequency bands with the image characteristics. For example, the regions of an image with intensities that vary slowly (e.g., walls in an image of a room) are characterized with intensities that vary slowly and are characterized with sinusoidal frequencies of low frequencies. On the other hand the edges of sharp objects are characterized by high frequencies. Hence, if we reduce the high-frequency components of an image will tend to blur it.

29.3 Median Filtering

Non-linear spatial filters are basing their processing on ordering of pixels contained in a region and replacing the center pixel with the value determined by the ranking result. The best known filter in this category is the median filter, which replaces the value of the center pixel by the value selected from the neighborhood of that pixel. Median filtering is an effective method for clearing the noise from an

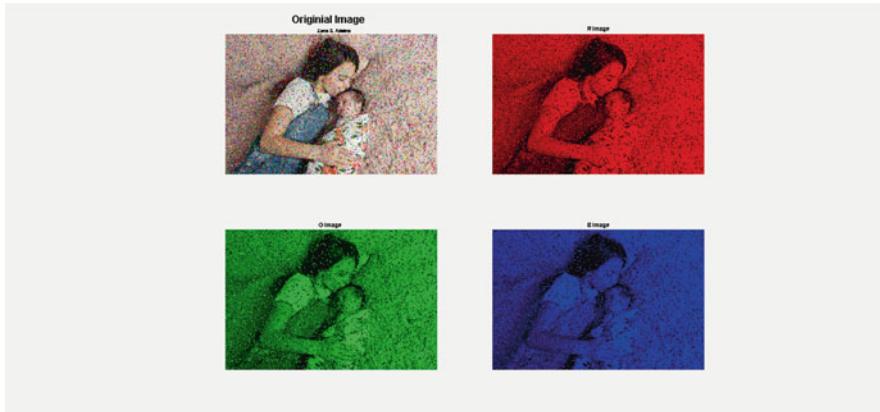


Fig. 29.13 Noisy and its decomposed (RGB) Image

image through non-linear filtering operation. Median filters provide excellent noise reduction capabilities for certain types of random noise, with considerably less blurring than linear smoothing filters of similar size. They are particularly effective in the presence of impulse noise (salt-and-pepper noise).

In order for this operation to be applied we need first to determine the size of windowing (e.g., filtering) operation that is going to be applied. Assuming we have determined the size of the window to be s_x and s_y and the process of filtering original image then can commence. The windowed image is applied by sorting the pixels that fall within the window in ascending or descending order. The median value of the sorted window is then picked as depicted in the following equation. The operation is repeated for every window within the image (Fig. 29.13).

$$-M \leq m \leq +M$$

$$y(i, j) = median(x(i - m, j - n))$$

$$-N \leq n \leq +N$$

Image after being cleaned by median filtering operation (Fig. 29.14).

29.4 Color Equalization

Digital images provide unique opportunity to manipulate the data for clarity of representation. The following examples make that point by demonstrating the enhancement procedure developed by the book author (Fig. 29.15):

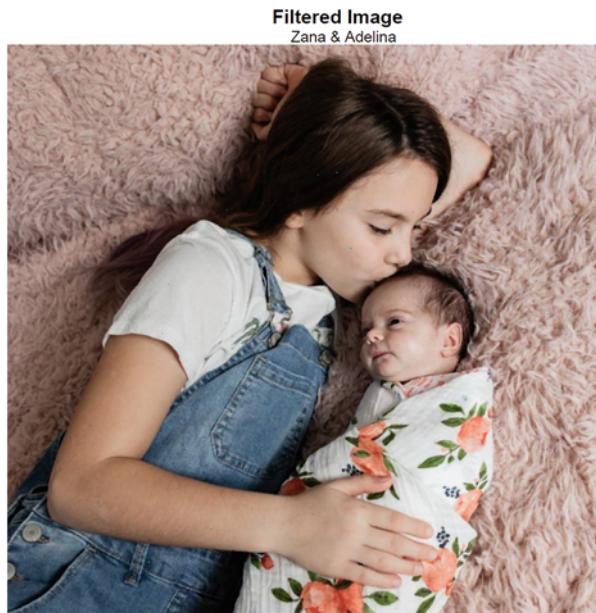


Fig. 29.14 Median filtered image

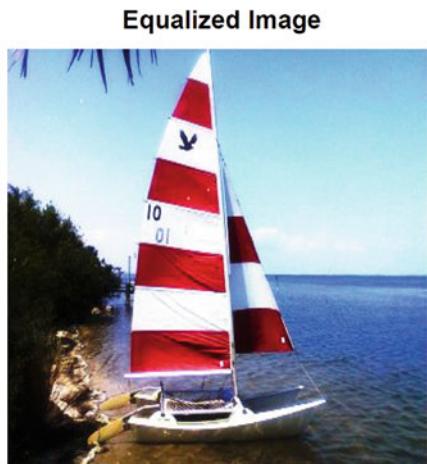


Fig. 29.15 Original and histogram Equalized Image

The process is going to be explained using only one channel, i.e., gray scale instead of colors—typically 3 channels like in the RGB, or any other color standard channel. Image is repetition of the same procure on each representation (e.g., RGB). Histogram Based Equalization of any gray-level modification technique is based on creating a mapping of the gray levels (or color) in the original image to the gray

levels in the modified image. Let \mathbf{x} represent the original image as a gray-level. Let \mathbf{y} represent the modified image.

$$\mathbf{y} = \mathbf{T}(\mathbf{x})$$

(1) **Histogram Computation:**

$$h(n) = \text{hist}(\text{image}(m,n))$$

(2) **Probability Distribution Function (PDF) of the histogram of the image of the size $M \times N$:**

$$pdf(n) = \frac{h(n)}{M \times N}$$

(3) **Cumulative Distribution Function (CDF) from the PDF:**

$$cdf(n) = I_{\max} \left(\sum_{r=0}^n pdf(r) \right)$$

(4) **Mapping of original image into transformed image:**

$$Eimage(m, n) = cdf(Iimage(m, n))$$

Example: Image of a boat (see figures presented in Figs. 29.16, 29.17, 29.18, 29.19, and 29.20).

29.4.1 Image Transformations

We have established earlier that images can be represented with a two-dimensional matrices, hence Image Transformation can be done via Matrix manipulations. Matrixes transform vectors from one space to the other. In geometry, an affine transformation or affine map (from the Latin, *affinis*, “connected with”) between

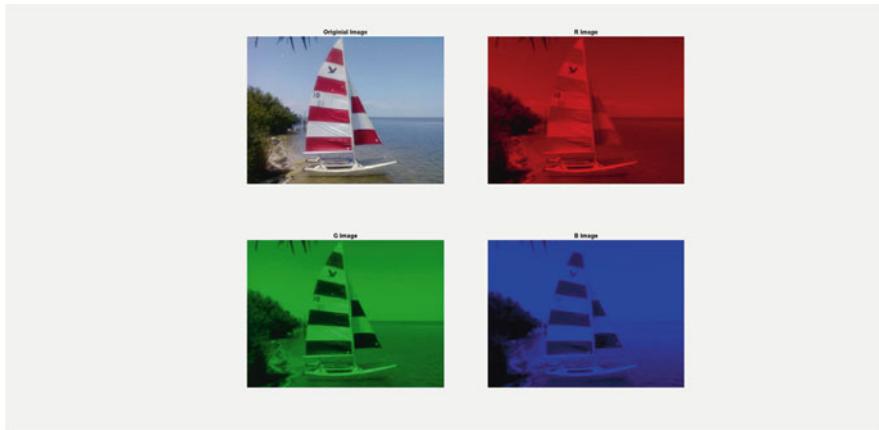


Fig. 29.16 Original Image and its Decomposed versions of Red, Green and Blue (RGB) channel

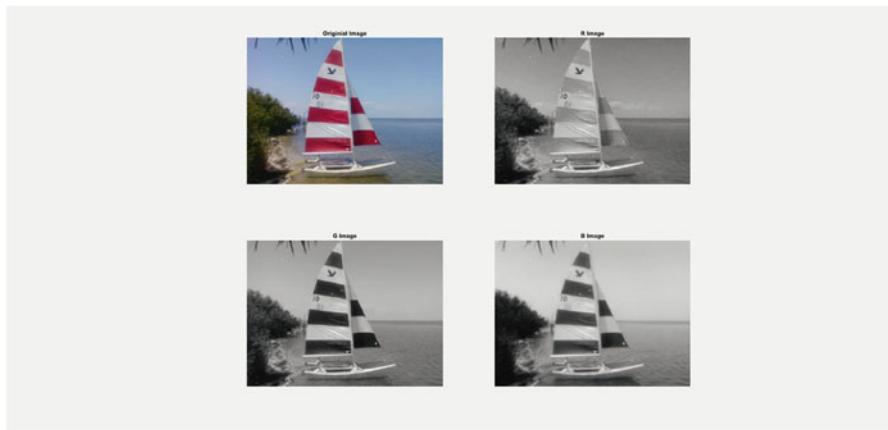


Fig. 29.17 Original Image and Decomposed Images versions of Red, Green and Blue channel presented in Gray scale

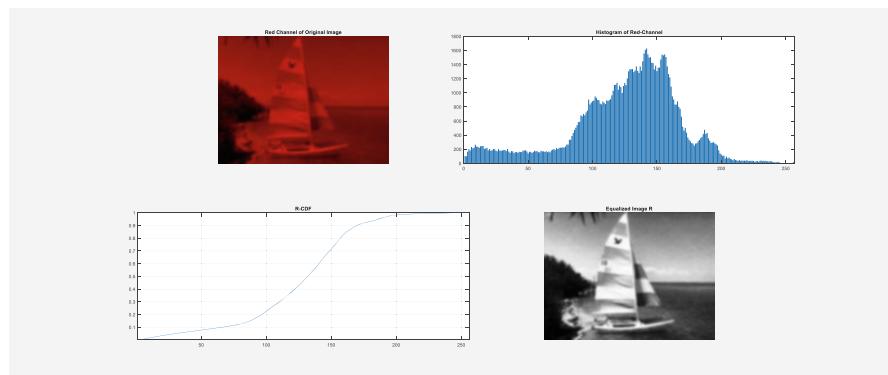


Fig. 29.18 Original and “Red” Channel Equalized Image

two vector spaces (strictly speaking, two affine spaces) consists of a linear transformation followed by a translation:

$$\mathbf{x} \Rightarrow \mathbf{y}$$

$$\mathbf{y} = \mathbf{Ax} + \mathbf{b}$$

This operation can be simplified by adopting the following notation for a generalized transformation matrix \mathbf{T} and extension of input vector \mathbf{x}' :

$$\mathbf{T} = \begin{bmatrix} \mathbf{A} & \cdots & \mathbf{b} \\ \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

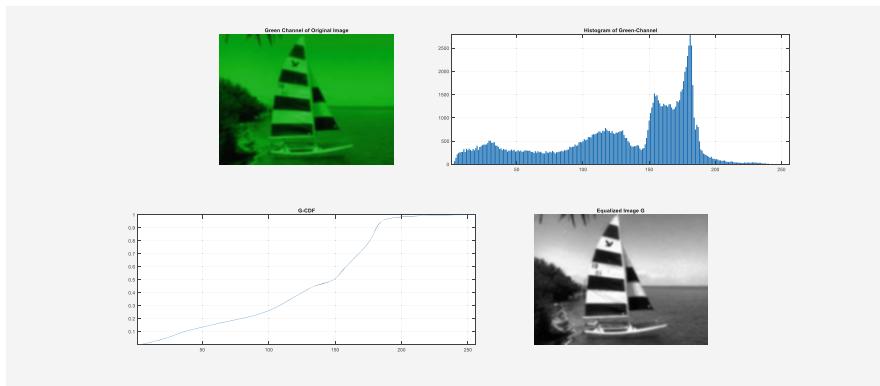


Fig. 29.19 Original and “Green” Channel Equalized Image

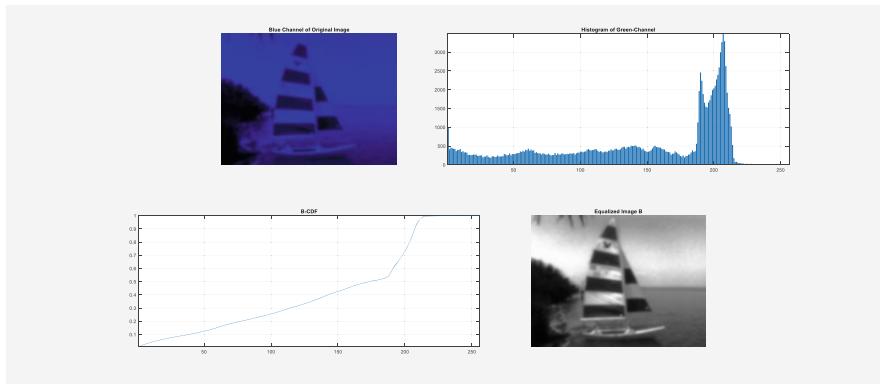


Fig. 29.20 Original and “Blue” Channel Equalized Image

$$\mathbf{x}' = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

This form is called homogenous coordinate system representation.

Image Transpose

The transpose image $\mathbf{B}(M \times N)$ of $\mathbf{A}(N \times M)$ can be obtained as $\mathbf{B}(j, i) = \mathbf{A}(i, j)$ or in matlab $\mathbf{B} = \mathbf{A}'$ (Fig. 29.21);

Matlab code used to generate transform of an image:

MATLAB Script:

```
function transposeImg(filename, type)
figure;
oimg = imread(filename, type);
subplot(1,2,1)
```

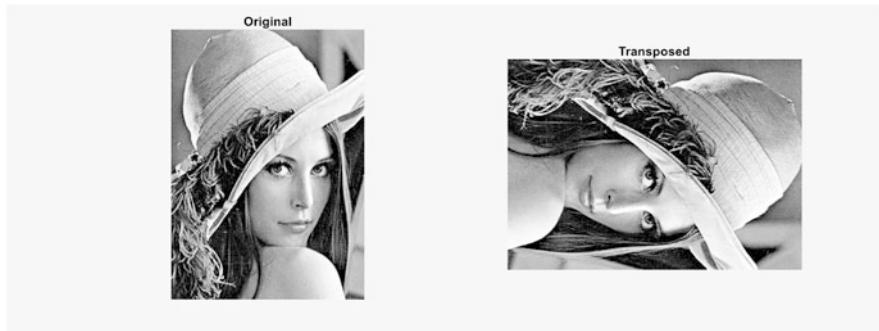


Fig. 29.21 Image of “lena” and its transpose



Fig. 29.22 Image of “lena” and its flipped version

```
imshow(oimg);
title('Original');axis off;axis image;
subplot(1,2,2)
J = imrotate(oimg,-90);
imshow(J);
title('Transposed');axis off;axis image;
```

Image Flip

The vertical flipped image $B(N, M)$ of $A(N, M)$ can be obtained as $B(i, M - 1 - j) = A(i, j)$: Matlab code used to generate transform of an image (Figs. 29.22 and 29.23):

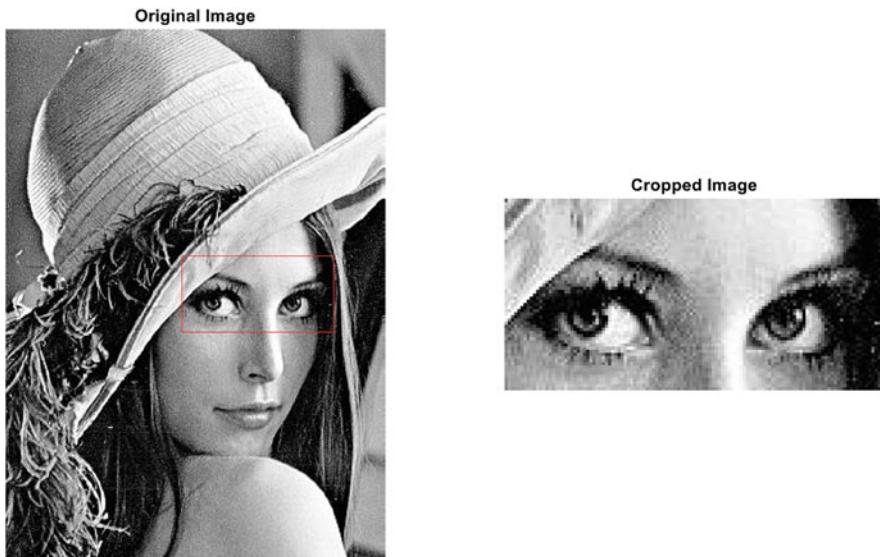


Fig. 29.23 Copped image from the original

MATLAB Script:

```
function flipImg(filename, type)
figure;
oimg = imread(filename, type);
subplot(1,2,1)
imshow(oimg);
title('Original');axis off;axis image;
subplot(1,2,2)
J = imrotate(oimg,-90);
imshow(J);
title('Transposed');axis off;axis image;
```

Image Crop

The cropped image $B(N_1, N_2)$ of $A(N, M)$, starting from (n_1, n_2) , can be obtained as $B(k, l) = A(n_1 + k, n_2 + l)$ ($k = 0, \dots, N_1$, and $l = 0, \dots, N_2$).

MATLAB Script:

```
function croppImg(filename, type, xmin, ymin, width, height)
I1 = imread(filename, type);
info = imfinfo(filename, type);
I2 = imcrop(I1,[xmin ymin width height]);
subplot(1,2,1)
imshow(I1); hold on;
plot([xmin, xmin+width], [ymin, ymin], 'r');
plot([xmin+width, xmin+width],[ymin, ymin+height], 'r');
```

```
plot([xmin, xmin+width], [ymin+height, ymin+height], 'r');
plot([xmin, xmin], [ymin, ymin+height], 'r');
title('Original Image');
subplot(1,2,2);
imshow(I2);
title('Cropped Image');
```

29.4.2 Examples of Image Transformation Matrixes

Translation (Fig. 29.24):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

⇓

$$x' = x + t_x$$

$$y' = y + t_y$$



Fig. 29.24 Example of mage “translation”

Scaling:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

⇓

$$x' = S_x x$$

$$y' = S_y y$$

Resizing:

The MATLAB statement $B = imresize(A, scale)$ returns image B that is scale times the size of original image A . If scale is in the range $[0, 1]$, the resulting image B is smaller than input image A . If scale is greater than 1, the resulting image B is larger than input A . By default, imresize uses bicubic interpolation. Example (Fig. 29.25):

Rotation

Some transformation is more difficult to accomplish than the other. Rotation transformation is one that clearly is more difficult to accomplish. Rotation transformation is accomplish with the following formula:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

⇓

$$x' = x \cos(\theta) - y \sin(\theta)$$

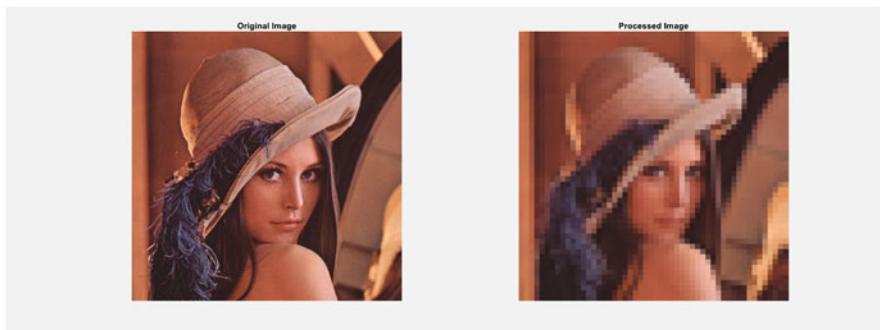


Fig. 29.25 Example of the resized image

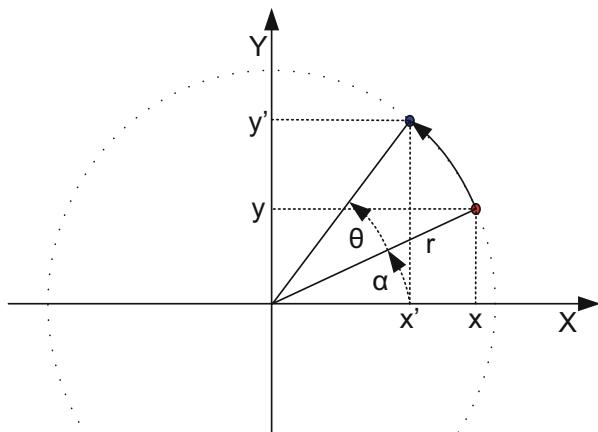


Fig. 29.26 Rotation of a pixel around an arbitrary origin by an angle θ

$$y' = x \cos(\theta) + y \sin(\theta)$$

The rotation of a two dimensional image is accomplished by utilizing the following graph (Fig. 29.26):

$$x = r \cos(\alpha)$$

$$y = r \sin(\alpha)$$

$$x' = r \cos(\alpha + \theta)$$

$$y' = r \sin(\alpha + \theta)$$

A useful trigonometric formula that applies here is:

$$\cos(\alpha + \theta) = \cos(\alpha) \cos(\theta) - \sin(\alpha) \sin(\theta)$$

$$\sin(\alpha + \theta) = \sin(\alpha) \cos(\theta) + \cos(\alpha) \sin(\theta)$$

⇓

$$x' = r \cos(\alpha) \cos(\theta) - r \sin(\alpha) \sin(\theta)$$

$$x' = x \cos(\theta) - y \sin(\theta)$$

&

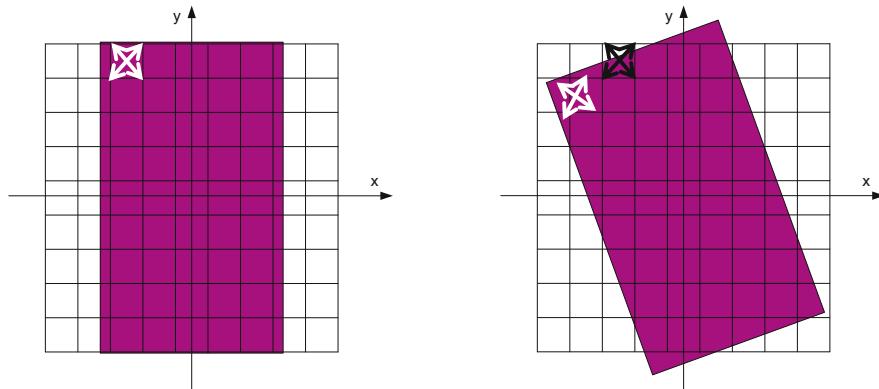


Fig. 29.27 Discrete nature of a digital picture can create problems with some transformations if not handled properly

$$y' = y \cos(\theta) + x \sin(\theta)$$

$$y' = x \sin(\theta) + y \cos(\theta)$$

Simple geometric transformations like translations, rotations and torsion's are not trivial because of discrete nature of the pixels as depicted in the next exaggerated picture (Fig. 29.27):

Example of rotation transformation as performed by MATLAB script:
function rotateImg(filename, type, angle)

```
figure;
oimg = imread(filename, type);

subplot(1,2,1)
imshow(oimg);
title('Original');axis off;axis image;

subplot(1,2,2)
rimg = imrotate(oimg,angle);
imshow(rimg);
title('Rotated');axis off;axis image;
```

29.5 Basic Image Statistics

Let S be a set, and define $\#S$ to be the cardinality of this set; i.e., $\#S$ the number of elements of S . The histogram $\mathbf{h}_A(l) = \#\{(i, j) | A(i, j) = l, i = 0, \dots, N - 1; j = 0, \dots, M - 1\}$ Where (Fig. 29.28):

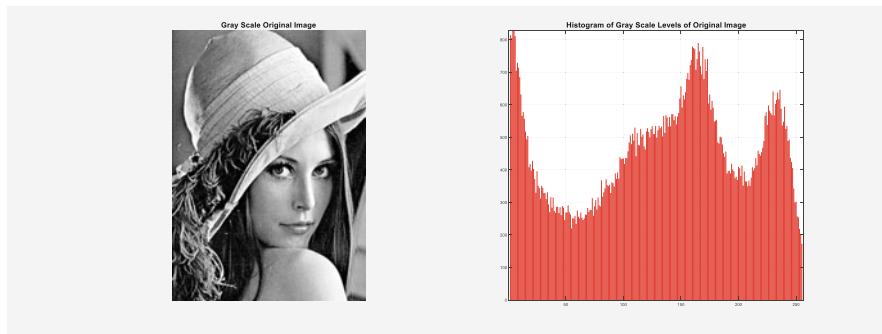


Fig. 29.28 Histogram of a image of ‘lena’

$$\sum_{l=0}^L h_A(l) = \text{Number of Pixels in } A$$

```

function histImg(filename, type)
%
% filename - inputfile
% type - jpg, tiff, etc.
%
figure('Name','Histogram Computation');
subplot(1,2,1);
orig_img = imread(filename, type);
imshow(orig_img);
title('Gray Scale Original Image');
%
% Compute Histogram
%
subplot(1,2,2);
edges = (0:1.0:255);
nbins = 256;
hgram = histogram(orig_img, ...
nbins, ...
'EdgeColor','r', ...
'BinEdges', edges);
title('Histogram of Gray Scale Levels of Original Image');
grid on; axis tight;
%
% Compute Histogram
%

```

```

figure('Name','Old Histogram Computation');
subplot(1,2,1);
imshow(orig_img);
title('Gray Scale Original Image');
%
% Compute Histogram
%
[m, n] = size(orig_img);
vec = reshape(orig_img, m*n,1);
hgram = hist(vec,[0:255]);
subplot(1,2,2);
% this is required due to buggy computation of hist
hgram(1) = 0;
hgram(256) = 0;
bar(hgram, 'LineWidth', 2, 'FaceColor', 'Red');
title('Histogram of Gray Scale Levels of Original Image');
grid on; axis tight;

```

Simple image processing techniques like transposing, flipping and cropping is being presented. Simple image statistics like sample mean and sample variance. Histograms are also presented.

- (1) $h_A(l)$: number of pixels in image A that have the value l .
- (2) Histograms tell us how the values of individual pixels in an image are “distributed”.
- (3) Two different images may have the same histogram

29.6 Abstraction Levels of Images and Its Representations

Depicted examples present various techniques developed to achieve a specific goal of making images clearer for human consumption. There a number of levels that this can occur starting from image transformation (image to image) ending with the most abstract level of image description (image to verbal description) as presented in Fig. 29.29. The objective of image understanding is a symbolic representation of the contents of an image. Applications of image understanding are computer vision and robotics. In this chapter, we have provided examples and applications of image enhancement, restoration and image understanding for image information retrieval.

Abstraction levels occur frequently in signal processes for defining properties of interest. This is also true for images. We see an overall organization for images in Fig. 29.29. One has to provide a computation that realizes each abstraction step. The organization starts from the lowest level, namely digital signals as described previously. For images received and stored by a computer it is the level of pixels. First, we provide a diagram illustrating the level structure. Figure 29.29 shows the abstraction levels on which an image is located. Humans just realize the very high

Information	Level	Examples/structure	Level of abstraction
Hidden	Pixel Level	2-dimensional image is defined as an array of $M \times N$ pixels.	Low
	Geometric Level	Visual properties of a image such as hue, brightness, shadings, texture, etc.	
	Symbolic Level		
Explicit	Level of the description		High

Fig. 29.29 Abstraction levels of an Image

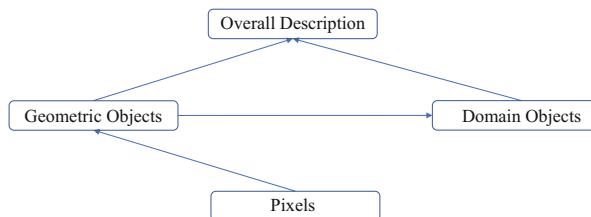


Fig. 29.30 Descriptive elements of an Image

levels. A machine, however, starts with the signal level and has to be told what the other levels are and how one can get to them. In general, we see the objects are shortly described in Fig. 27.2. This diagram illustrates much of the hierarchy used for image interpretation (Fig. 29.30).

We describe in short the content and principle structure of the levels. Here we deal with a multivariate structure. One distinguishes between two and three dimensions.

29.6.1 Lowest Level

The lowest level is the pixel level; the signals are called voxels. In two dimensions the structure is defined as an array of $M \times N$ (e.g. 1024×1024) pixels where a pixel is e.g. a gray value, a color value or an element in the range $\{0, 1\}$ (black, white). The notation for the pixels we take the form a_{ik} . In three dimensions the structure is

defined as an array of $M \times N \times L$ (e.g. $1024 \times 1024 \times 64$). Hence a voxel is of the form a_{ijk} .

29.6.2 *Geometric Level*

The next higher level is the geometric level. Here elementary geometric objects are introduced as lines and curves, areas with their boundaries and brightness, segments etc. In addition, this level shall also deal with shades, texture information and related things. One has to take into account that real objects do not occur as exactly mathematical objects and they do not satisfy precisely the mathematical definition. In addition, some of them are vaguely defined as e.g. “ovals” which are informal versions of ellipses.

29.6.3 *Domain Level*

On top of this level we find the domain level. Here symbols that people realize visually are introduced. They are either constructed from the pixel level by image processing methods or defined symbolically in terms of geometric objects. From our point of view, we will not be concerned with this level and we do not deal with the overall level. A simple example: Schematic image of a house as shown in Fig. 29.31.

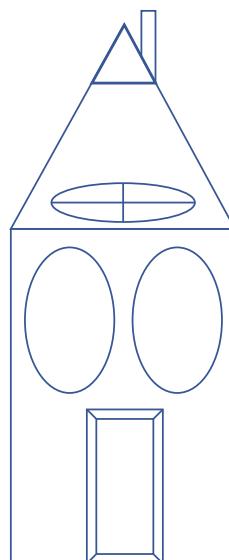


Fig. 29.31 Sketch image of a house

This image uses just three geometric forms, rectangles, triangles and ellipses. If there are finitely many and simple forms one can easily store them in a data base.

29.6.4 Segmentation

For signal processes, segmentations play a big role. There the idea of segmentation is to partition the process into smaller parts. Function of the application, a reason can that smaller segments are easier to handle, that the segments have some better internal similarity, or that that they bear special functions. This is oriented on a linear structure of the signals and a partition generates smaller intervals. Segmentation is frequently used for multivariate systems. In an image, one also generates sub images with fewer pixels. Here the idea is to do this in such a way that the pixel values in a segment are similar to each other. The similarity is measured by a homogeneity predicate HP; the definition depends on the specific application. A simple example is $\max(S) - \min(S) > t$ for some t where max and min functions are taken for the gray values of the pixels from the region S .

- One calls then the regions of the partition homogenous with respect to HP.
- There are no specific geometric forms of the regions required.
- HP may contain parameters with the effect that they put increasing conditions on the homogenous regions which results in a refinement of the regions.

A segmentation algorithm can fail in two ways:

- Under segmentation: Two objects are incorrectly merged into one object.
- Over segmentation: An object is incorrectly divided into parts.

There is no absolute definition of a segmentation being correct. It refers to the fact that the resulting image reflects the real object correctly. This is beyond the scope of image processing methods and requires access to a knowledge base. This is quite the same as for other applications like speech. Segmentation is important for image understanding because it can lead to distinguish objects; the individual objects are often homogeneous and the joining of different objects is not.

29.7 Background Information

Multimedia have been discussed in detail in Vaseghi (2007). In addition, this book contains very much interesting material on general stochastic processes. Multivariate systems and filter banks have played a role in source coding and compression for contemporary communication applications. In multivariate systems one can represent highly complex objects that are of practical interest. One distinguishes between one dimensional and multidimensional multivariate systems.

Several applications such as conversion between video signals require the use of multidimensional multivariate systems.

Past

In multidimensional multivariate systems, the basic techniques include multidimensional digital filters. See Tsuyan and Vaidyanathan (1993) and Suter (1997). For general multivariate systems see Vaidyanathan (1993). Images are in some sense simple systems. They are given as pixel matrices and the only relation between them is the neighborhood. However, here the problem area starts: One can express arbitrary complex objects in terms of images. There is a huge literature on the creation and reception of them.

Suggested

Many different journals provide state-of-the-art resource content on biomedical imaging including IEEE Transactions on Image Processing, IEEE Transactions on Information Technology in Biomedicine, Journal of Biomedical Optics, Journal of Computer-Assisted Microscopy, Journal of the Optical Society of America, Analytical and Quantitative Cytometry and Histology, and Cytometry. Image understanding is quite different from image processing, Image processing can be done without knowing from which area the image is. Images cannot be understood without knowing this and it can even for humans sometimes be impossible. For image understanding one can refer to Richter and Weber (2013). Abstraction levels for images are more specific than for general signal processes. See Kung and Taur (1995), For image processing (Blancet and Charbit 1998; Igarashi et al. 1999), are useful. Neural nets have also been used successfully in this area. For instance one can incorporate directly in an image the pixels and their associated neighborhoods in the network. For fuzzy neural nets see Kosko (1992).

Readers are encouraged to read references such as Gonzales et al. (2018), Kung and Taur (1995), and Harb and Husseiny (2000) for details, that are used in this chapter among other many references.

29.8 Exercises

Exercise 1 MATLAB provides a wealth of tools that can be applied for image processing. Among functions are: [counts,binLocations] = imhist(I), J = histeq(I,hgram), J = adapthisteq(I), J = imadjust(I), etc. Describe how you can use the tool imtool to enhance an image (e.g.,imtool('cameraman.tif')).

Exercise 2 Describe a transformation from images of houses to text and describe in particular which attributes you need.

Exercise 3 Describe a picture of a house purely in terms of the elementary signals.

Exercise 4 Take a medical picture of your choice and describe it with respect to uncertainties.

References

- [Gonzales2018] Gonzales, Rafael C., and Woods, Richard E., Digital Image Processing, (2018), 4th Ed., Pearson.
- [Kung1995] Kung, S.Y., Taur, S, T, (1995), [Kun], For image processing G. Blanchet, G., Charbit, M. (1998), [Bla] and Igarashi, T., Matsuoka, S., Tanaka, H. (1999)
- [Tsuyan1993] Tsuyan, C., Vaidyanathan P.P. (1993).Recent Developments in Multidimensional Multivariate Systems. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3.1993. Suter, B.W. (1997).Multivariate and Wavelet Signal Processing. Elsevier Science.
- [Blancet1998] Blanchet, G., Charbit, M.(1998). Digital signal and image processing using Matlab, Springer-Verlag
- [Vaidyanathan1993] Vaidyanathan, P.P. (1993). Multivariate Systems and Filter Banks. Prentice Hall, N.J., USA
- [Igarashi1999] Igarashi, T., Matsuoka, S., Tanaka, H. (1999). A sketching interface for 3D freeform design. 26th annual conference on computer graphics and interactive techniques. ACM Siggraph 99.
- [Richter2013] Richter, M.M., Weber, O. (2013). Case-Based Reasoning. Springer Verlag.
- [Kung1995] Kung., S.Y., Taur, S, T, (1995), Decision-based neural networks with signal/image classification applications, IEEE Trans. Neural Networks, vol. 6.
- [Kosko1992] Kosko, B. (1992), Neural Networks and Fuzzy Systems, Englewood Cliffs, NJ, Prentice-Hall.
- [HarbandHusseiny2000] Harb, H., A. H. Husseiny (2000), [Ha]. Isolated words recognition using neural networks. International Conference on Electronics, Circuits and Systems, 1:349–351.
- [Vaseghi2007] Vaseghi , S.V (2007) Multimedia Signal Processing, John Wiley & Sons Ltd, 2007.
- [Suter1997] Suter, B.W., The multilayer perceptron as an approximation to a Bayes optimal discriminant function, IEEE transactions on neural networks 1(4), 291, 1990

Chapter 30

Visualizations of Emergency Operation Centre



Overview

The emergency operation centre (EOC) is the place where the incident management system is run, one collects information, makes decision about the priorities, and coordinates actions and communications. Communications is one of the most critical stage during emergency situations. Communication semiotic and can take different forms. The EOC plays a crucial role in managing emergency situations and relates people, systems and things. This chapter is a glimpse of visualization and visual story telling and how these help to communicate, maintain situational awareness, location awareness, and detect events in complex situations.

30.1 Introduction

Communications, systematic presentation of scenario and the background using visualization, handling emergency situations optimally is discussed broadly in machine learning. Visualization can be a useful tool to manage emergency situation optimally and to locate negligence or inefficient responses to the emergency situations. Potentially interesting patterns can be identified through information visualization. This in fact assists in discovering relationship in the patterns which may eventually lead to find out the facts. Visual perception of the data reveals the insight of the data better than describing the scenario by texts or verbal expressions. The data used in emergency operation centre generally come from various sources, and different users, who need different knowledge, using adequate and different images to tell the visual story narrate the scenario and context vividly. One objective of the EOC is to identify potentially interesting patterns, discover the relationship in the patterns. Visualization help in perceiving the insight of the data easily.

Information visualization helps think, assist humans in solving problems, unveils underlying structures. An example of a meaningful use of visualization, is an

emergency situation. Emergency situations and the need for rapid, accurate communication for informed action by management, first responder, and the public bring to the forefront the need for analysis to effectively communicate what they know. By nature, data are varying quality, and most data have levels of uncertainty associated with them. As data processes, and transforms, the data uncertainties most likely intensified. These uncertainties may have profound effects on the analytical process and must be portrayed to users to inform their thinking. The users will make their own judgements of data quality, uncertainty, and reliability, based upon their expertise. These judgements must be captured and incorporated as well. Furthermore, in this constant change, assessments of data quality or uncertainty may be called into questions or any time based on the existence of new and conflicting information. The complexity of this problem will require algorithmic advances to address the establishment and maintenance of uncertainty measures at varying levels of data abstraction.

30.2 Communications in Emergency Situations

The problem is that the people who support emergencies are often unaware of what has happened. They might be in their offices or similar locations and need to be informed. The information should enable them to act properly where effective communication plays a crucial role. Communication can take place in various ways. Communication needs modes where the participants exchange messages. Here we use a visual way because the participants are not together and writing is too lengthy and time consuming. In images one can in principle tell a whole story that leaves the opportunity to act based on a partially own decision. This allows to use own equipment optimally. Risk information was visualized properly in Fig. 30.1 by Eide (2012) in “Geographic visualization of risk as decision support in emergency situations” to meet the need and get insights for right emergency management solutions.

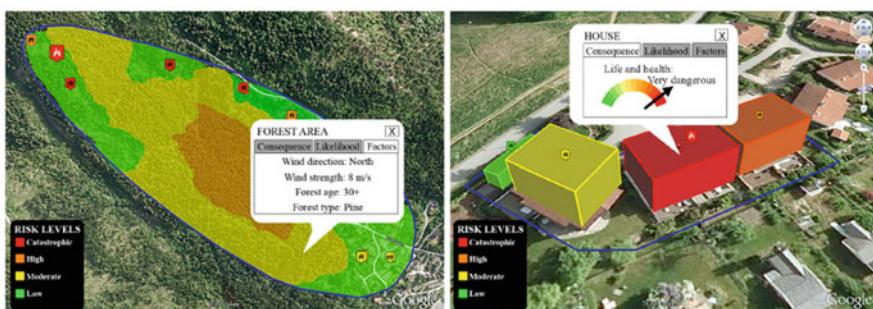


Fig. 30.1 Risk visualization example in emergency situation (Eide 2012)

30.3 Emergency Scenario

In a typical emergency situations have collapsed buildings, break outs, nature of injuries, location of the injuries, damaged evacuation of an area, various dimensional disasters such as shooting, earthquake, rubble caused, bombing, city fire, live damages, accidental injuries, and deaths or a hospital to rescue the injuries, large building for rehabilitation, possible signal, sign, and barricade, if an area of building appears to be unsafe, begin evacuation procedure. The EOC deals with all such emergencies. Persons, buildings, or houses, cars are some example of objects used in EOC. These objects are defined by several classes.

30.3.1 *Classification and EOC Scenario*

The visualization uses images. An abstract complex image can tell a story. The story does not only present details, but also allows the users to make own decisions for support. Then the question is how to generate the images in such a way that they represent complex interesting situations that are understandable to humans. Basically an image is a set of pixels. To a human a set of pixels says almost nothing. The meaning is obtained at higher levels of abstraction in such visualization. The first topic is to apply feature extraction to the classes. The classes determine as a consequence the images. The level of abstraction for images can be the lowest level, geometric level, individual objects, or the overall situation. Here we will uses the image in the data at the highest level.

The visualization needs to employ all levels. It refers to classification. When the top level class is provided then the details of the higher levels have to be determined. The geometric level can be uniform to all applications, probably except some elements of higher dimensions. All higher levels depend on the classes. However, several classes use the same objects. The most common objects used by several classes are:

Classification is a structural concept that uses classes to the given data. There are a number of ways to classify them. In this situation, several classifications in a hierarchical way are needed. This is briefly introduced

Upper level classification: On the upper level we consider different types of emergencies. These are:

- Fire
- Earthquakes
- Bridge Collapsed
- EQ Effect1
- Liquefaction
- Outbreaks
- Live Damaged

These classes are vocabulary data that belong to each emergency type. These classes are not disjoint. For instance, the object house can belong to fire as well as to traffic accidents.

Subclasses:

For each classes there are a number of subclasses of each class, others only of specific ones. **Subclasses of all classes:**

- Location
- Buildings
- Emergency Time
- Number of Persons involved (Roles)

Subclasses of specific classes:

- Types of live damaged
- Types of Buildings
- Types of fire
- Types of cars
- Roadblocks
- Rubble caused by incidents

30.4 Technical Aspects and Techniques

Machine learning namely supervised, unsupervised, or reinforcement learning and included techniques such as regression, classification, and clustering, data visualization, visual data mining are some standard methods and techniques for EOC data analysis.

30.4.1 Classification

The clustering methods are used for identifying classes. They have sometimes a different character depending on the classes, the communication is needed to be clarified. The data have to be interpreted to the user in order to react properly. The communication is difficult, because there are different types with different requirements. The classes support the principal character. The purpose of classes is to deal with the emergency situation properly, and this is making use of classes. The upper level of the classes which has to be informed.

30.4.2 Clustering

In the terminology of learning, clustering techniques fall into unsupervised learning. In this EOC context, one wants to learn the mentioned classes above from the given data. Classes and clusters are sets of objects. When clusters are estimated, they could agree with the classes and learning wants to achieve this. Therefore one applies clustering for learning. Clustering assumes some distance or similarity measure and one wants that in this respect the elements are close to each other. In general clustering methods may produce an arbitrary number of clusters. K-means is a common clustering technique. The number of clusters is always constant to K. That means that the number of classes is also K and has to be known in advance. But K-means is not sufficient to many clustering problems. Such complex real world problems will be accommodated in our next endeavour.

More information about prediction, clustering and classification relating to visualizations, and emergency situations will be detailed covered later.

30.5 Background Information

An emergency situation and its consequences can be varied to a different extent. Disasters cause damages to human lives, properties, and infrastructures. Natural disasters may include risk from disease, floods, hurricanes, tornadoes, earthquakes and wildfire. Examples of man made disaster can be terror attacks, shootings, spills or toxic waste hazards. Natural disasters such as seismography, earthquakes, volcanoes negatively affect society through damages to property loss, loss of life and can be beyond.

The data used in emergency operation centre generally come from various sources and different users who might be interested in different kinds of knowledge. Techniques in data mining in combination with statistics, visualization are used to find the knowledge in the data, to discover interesting patterns, associations, correlations, searching for interesting patterns that are frequent. The data can be spatial attributes such as points, line and region, a non spatial attributes such as emergency events, roles, and actions. Spatial auto-correlation such as Moore I is a preliminary method that is used in spatial clustering. Association rule mining a priori associations, correlations and dependencies among roles, events, and actions.

Data visualization, and visual analytics help in emergency management, identifying next countermeasure, observe slowly varying and changes in global warning, and environmental changes by disasters that faces major catastrophes. Visual analytics has been widely applied in defence and security related areas and emergency services.

An EOC is responsible for strategic direction and operational decisions. The EOC collect, gather and analyze data; make decisions that protect life and property.

The staff of an EOC is properly trained to carry out actions and to responds to emergencies as needed.

Emergency is a situation that possess an immediate risk and danger to health, life, property, environment, loss of life. According to Wikipedia, an EOC is a central command and control facility for carrying out the principles of emergency preparedness, emergency management, and disaster management at a strategic level during an emergency, and ensuring the continuity of operation.

EOC was first introduced during Civil defense in United States. During 2000 to 2008, earthquakes were responsible for an average of 50,184 deaths and floods affected an average of 99 million people per year. Moreover, there are growing number of mortality due to terror attacks. There are many other incidents. The motivation of this chapter is related to a devastating Flood in Alberta, Canada in 2013 due to lack of improper EOC management.

EOC plays a significant role in emergency planning. Emergency and disaster need coordination, preparation, and process of uses of testing and updating. More information about visualization for emergency can be found in “Emergency Planning and Visualization: the case of Miami-Dade County’s Emergency Operations Center by A. Castellanos, A. Castillo, A. Gudi and R. Lee, by WIT press, 2013,”, “Visual Analytics: Scope and Challenges” by Daniel A. keim, Florian Mansmann, Joern Schneidewind, Jim Thomas and Harmut Ziegler, on the website <http://nvac.pnl.gov>. The reference has a list of recommended text for the readers. The visual story telling application in emergency situation is in progress and will be enhanced in our next endeavour. This chapter is an essence of an application where one can see how visualization is applied in an emergency operation centre at an abstract level.

Many references such as Refs. Zhang et al. (2010), Amicis et al. (2009), Huand and Huang (2013), Myatt (2007), Benz et al. (2010), PSC (2020), and EOC (2020) are used in this chapter and readers are encouraged to read these and similar topics for details.

30.6 Exercises

Exercise 1 Collect twitters tweets on disaster events, and visualize the natural disasters on the map, display all the disasters on the map double click on ‘Latitude’, ‘Longitude’, ‘Disaster Type’, and ‘Time’.

Exercise 2 Look at some tableau examples relating to graphs, maps, chart, and complex visualization example of disasters; write them your visualization and understandings about the graphs such as what is going on in the graph, what effect does this have on the environment.

References

- [**Eide2012**] Eide, A. W., Stolen, K., Geographic Visualization of Risk as Decision Support in Emergency Situations, Human System Interactions (HSI), IEEE, vol. 5, June, 2012
- [**Zhang2010**] Zhang, Q., Segall, R. S., and Cao, M., Visual Analytics and Interactive Technologies: Data, Text and Web Mining Applications, Publisher IGI Global, October, 2010-10-3 (bag of words, ontology mapping)
- [**Amicis2009**] Amicis, R. De, Stojanovic, R., and Conti, G., GeoSpatial Visual Analytics: Geographical Information Processing and Visual Analytics for Environmental Security, Publisher Springer, January, 2009
- [**Huand2013**] Huang, M. L., Huang, W., Innovative Approached of Data Visualization and Visual Analytics, IGI Global, July, 2013
- [**Myatt2007**] Myatt, G. J., Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining, John Wiley & Son Inc., 2007
- [**Benz2010**] Berrz, M.W., Kogan, J., Text Mining - Applications and Theory: Applications and Theory, John Wiley & Sons, Incorporated, February, 2010
- [**PSC2020**] National Emergency Response System, Public Safety Canada, <https://www.questia.com/library/politics-and-government/public-policy/emergency-disaster-management>, 2020
- [**EOC2020**] Emergency operations center, https://en.wikipedia.org/wiki/Emergency_operations_center, 2020

Chapter 31

Intelligent Interactive Communications



Overview

Communication is multi-modal. Human perception and everyday interaction is multi-modal. Modality is defined by its physical medium and a particular way of its representation. Multi-modality is a combination of diverse communicative forms led by transformation and interpretation. The goal is 'making sense'. Multi-modal communication combines multiple data sources for an effective and meaningful representation. A concept of multi-modal intelligent interactive machine in civilian and military communication is introduced in this chapter. This applies automation capabilities to communicate interactively applying verbal and non-verbal multi-modalities. The modes are speech, text, gesture, images, graphics and visualization.

31.1 Introduction

Communication, a fundamental human need, plays a crucial role in each aspect of human life. It is formed by meaning and is semiotic in that it uses speech, gaze, touch, images and on demand real or virtual assistance. Communication is multi-modal as human perception and everyday interaction are multi-modal. Modality is defined by its physical medium and a particular way of its representation. Multi-modality is a combination of diverse communicative forms led by transformation and interpretation. The goal is 'making sense'. We see a multimodal human robot communication in Fig. 31.1. We call this framework as intelligent interactive communication. In this framework, a multi-modal autonomous robot offers communication using speech, images, gestures, visualization and virtual environment onto emergency, tactical and operational platforms through a multi-modal user interface as shown in Fig. 31.2 for humans to interact with the Interactive Robot System (IRS). This is a multi-modal Human-Computer Interaction (HCI): An interdisciplinary endeavor.

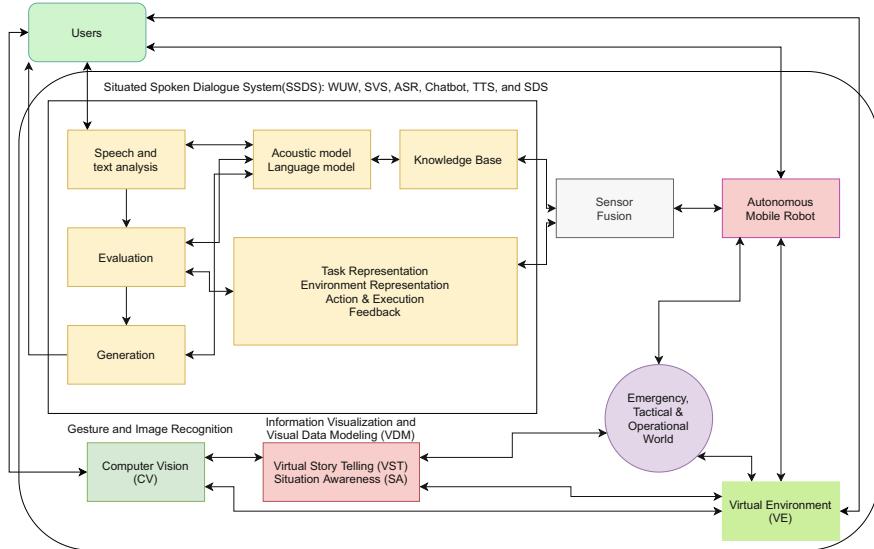


Fig. 31.1 Intelligent interactive machine for multimodal human-machine communication

When constructing an intelligent system with internal representations of symbolic nature, and with cognitive activity corresponding to computational manipulation of these symbolic representations, the effort is to define them consistently to the external world. With symbolic reasoning, the system knows what is in its knowledge base system as they are represented in a formal language, which is readable and understandable. Multimodal interfaces offer different modes of interaction, from visual to voice to touch, according to changes in context or user preference.

Multiple modes of communication are involved in the development of intelligent HCI based machine and multimodal user interface. The modes are:

- Speech
- Text
- Gestures
- Images
- Visualization
- Personal assistance in virtual environment

Figure 31.1 is the block diagram of the autonomous interactive robot for multimodal intelligent interactive communication. This is a complex machinery based on multimodal human machine interaction discipline. The user can select a suitable communication option using the multimodal interactive interface as shown in Fig. 31.2.

The robot control, including reactive, hybrid, and behavior-based control emerged as a result of lessons learned from control theory (the mathematics of controlling machines), cybernetics (the integration of sensing, action, and the

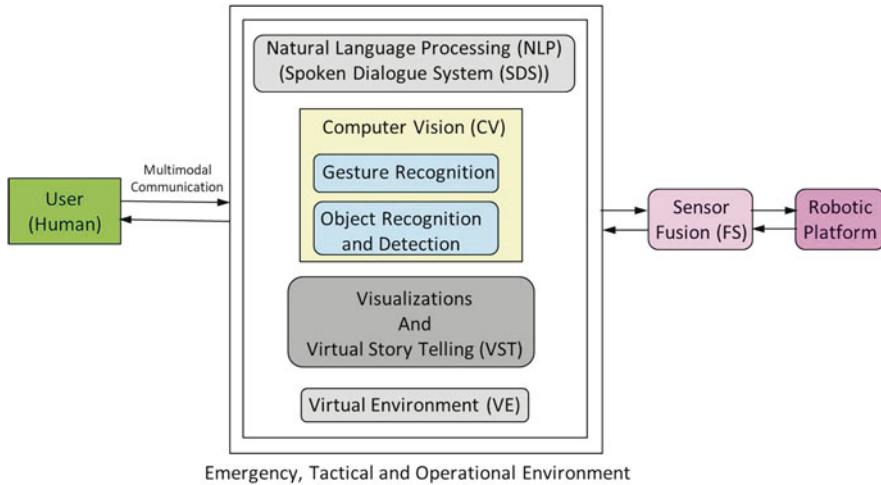


Fig. 31.2 Multimodal autonomous interactive interface

environment), and AI (the mechanisms for planning and reasoning). A robot is an autonomous system which exists in the physical world, can sense its environment, and can act on it to achieve some goals. Automated tracking of humans and moving objects using computer vision, and other sensors for thinking, acting, and the interaction with the environment.

Multimodality is an interdisciplinary endeavour. The challenge is to hold together throughout the central orientation to the phenomenon of multimodality regardless of various methods, disciplines and modes. The interdisciplinary artefacts in IRS is shown in Venn diagram in Fig. 31.3. These areas are robotics, AI for robotics, machine learning, Natural Language Processing, Visual Data Mining, computer vision and virtual environment.

31.2 Spoken Dialogue System

A spoken dialogue system (SDS) is a computer based system that enables a user to bilaterally communicate via spoken language with a machine (hardware and/or software). Three fundamental SDS components are:

- Acoustic front-end
- Semantic layer
- Logical layer

The suggested Situated Spoken Dialogue System has following five components shown in Fig. 31.4:

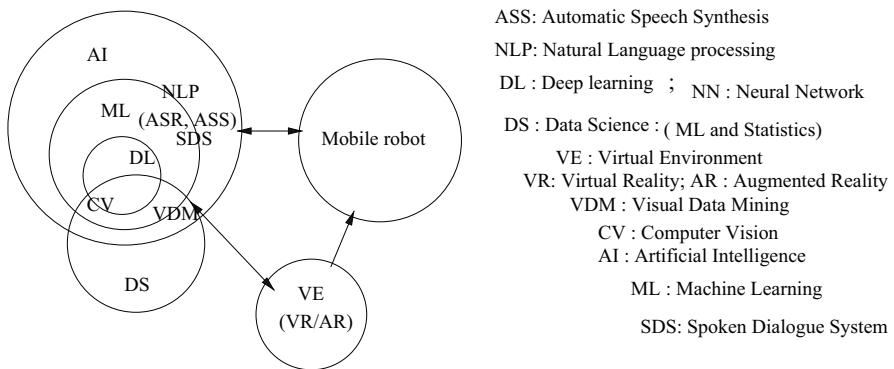


Fig. 31.3 Interdisciplinary intelligent interactive multimodal communication

- (1) Noise robust interactive environment via a microphone array
 - (2) Context-aware SDS
 - Wake-Up-Word (WUW): This activates the system using a trigger. It also detects the dialogue turns to maintain the flow and consistent natural short conversation in a dialogue form
 - Trigger based ASR: Confirmatory triggered words and sounds with visual symbols for the four different basic turns essential to these scenarios
 - User input for the ASR: Extension of WUW to SVS
 - (3) Chatbot: Interactive communication using Conversational AI tool called RASA
 - (4) Reporting functions of the robot through an interface between the human and robot. This has following components:
 - Automatic Speech Recognition: Among different speech recognition development tools, Kaldi speech recognition system tool is selected for this development. This is discussed details in <https://kaldi-asr.org>.
 - Text to Speech Synthesis: MaryTTS multilingual Text-to-Speech Synthesis and Mozilla TTS deep speech engine are some options for this.
 - Display for feedback, functionalities and reporting through speech and text.
- [1] Wake-Up-Word (WUW)
 [2] Chatbot for text based interaction
 [3] Speaker Verification System (SVS)
 [4] Spoken Dialogue System (SDS) for spoken dialogue based interaction

Interactive chatbot based on RASA is shown in Fig. 31.5.

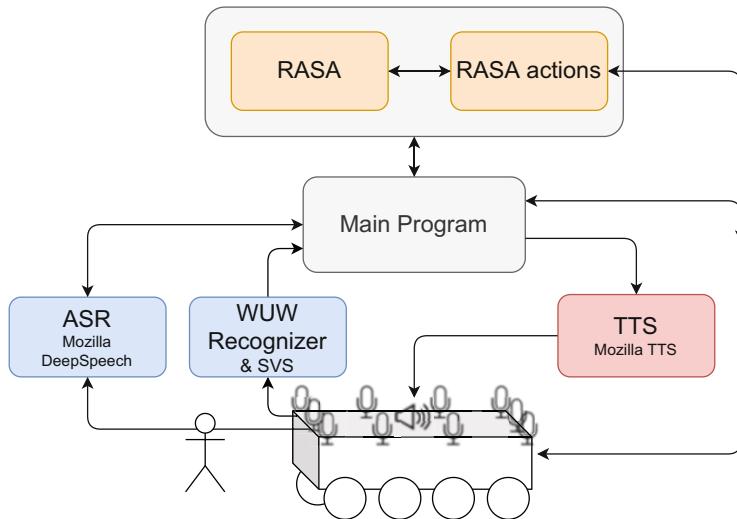
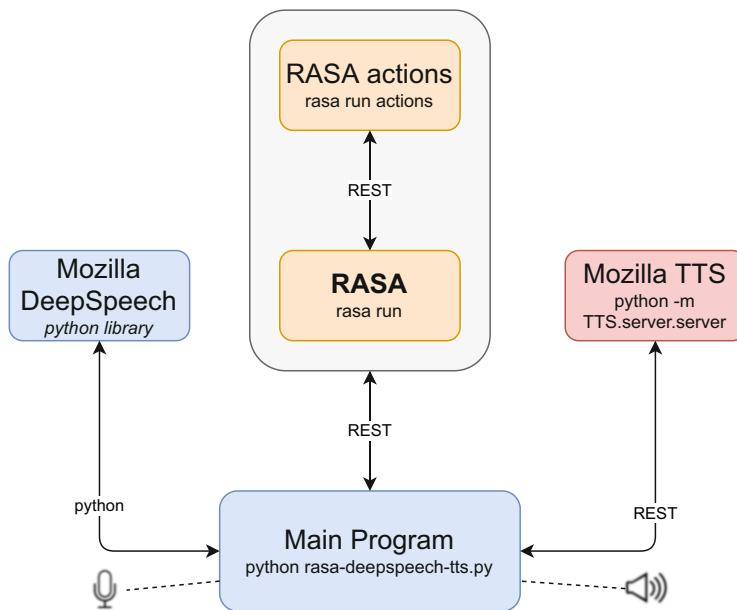


Fig. 31.4 SSDS components

Fig. 31.5 Chatbot for text based interaction using RASA <https://rasa.com>

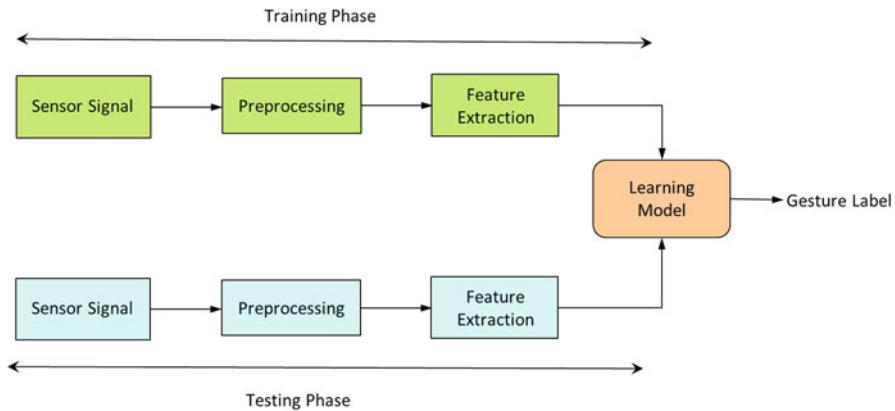


Fig. 31.6 Gesture recognition for communication

31.3 Gesture Based Interaction

Hand gesture can be classified into two categories: static and dynamic. A static gesture is a particular hand configuration or pose that is represented by a single image, while a dynamic gesture can be considered as a continuous motion of the hand. Gestures data are collected by sensors in both training phase and testing phase at pattern recognition building block shown in Fig. 31.6. The collected data are prepossessed for feature extraction. Example of gesture features are position of body joints, the shape of the hand, the angle. The features of the gestures are trained in the training phase using learning models and tested with the given data to classify the gestures in the testing phase. Gesture recognition research uses the joints of the human body to anchor positional data of ‘configurations’ or ‘positions’ of a human body at particular points in time. Gesture recognition is an interactive process between a user and a receiver. The user or receiver both can move hand in any direction at any angle in all vector spaces. How the gesture will be used to interact in different situation will be explained in details through case studies in our next book.

31.4 Object Recognition and Identification

Object recognition is an application of machine learning and deep learning. This identifies the object present in images and videos by recognizing the content of an image. Figure 31.7 shows a standard approach of object recognition. The hypotheses formation and object verification are managed via feature classification, feature matching and model.

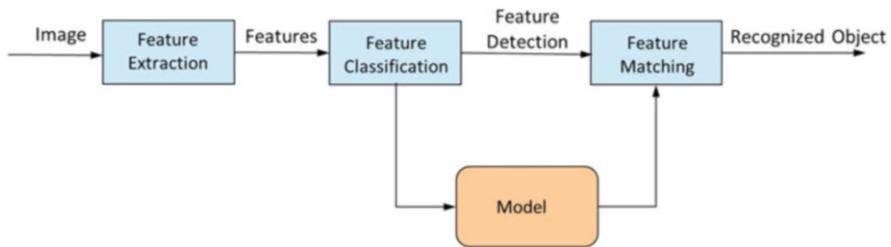


Fig. 31.7 Object detection and recognition (Jain and Kasturi 1995)

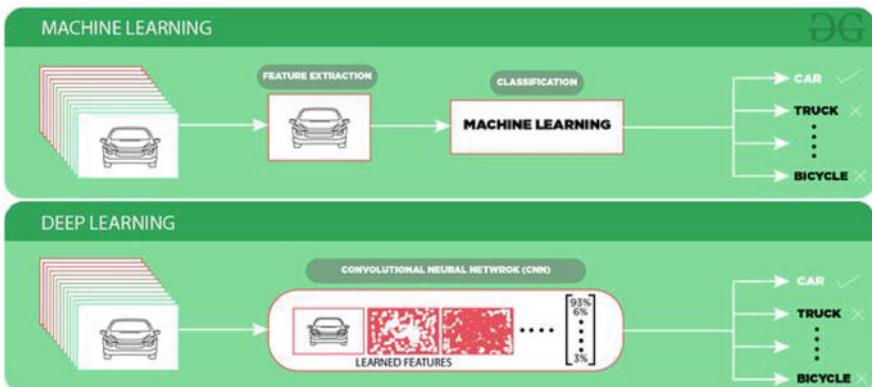


Fig. 31.8 Object detection and recognition <https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/>

Convolution Neural Network (CNN) takes an image as input and outputs the probability of the different classes. It is widely used and most state-of-the-art neural networks used this method for various object recognition related tasks such as image classification.

Object Detection algorithms act as a combination of image classification and object localization. It takes an image as input and produces one or more bounding boxes with the class label attached to each bounding box. Image classification takes an image as an input and outputs the classification label of that image with some metric (probability, loss, accuracy, etc.) (Fig. 31.8).

31.5 Visual Story Telling

The Visual Story Telling (VST) as a visual media in multi-modal communication assists in visualization, cognition and perception to prediction. Basic VST building block shown in Fig. 31.9 includes information processing. This is extended to pre-

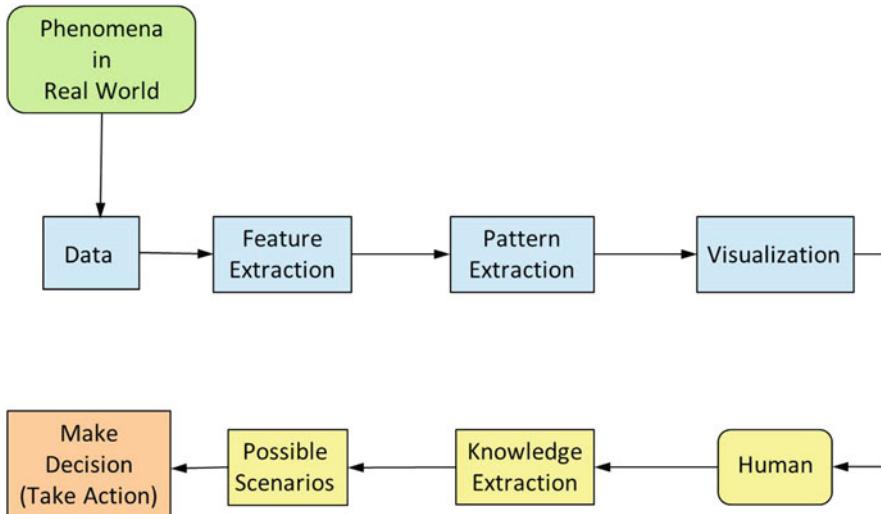


Fig. 31.9 Visualization

dictions and descriptions shown in Fig. 31.9. This includes information processing through VDM. Thus the VST captures abstract data, encodes data for visualization, transforms them for cognition and perception to discover knowledge and predict hypothesis as shown in Figs. 31.9 and 31.10. Analytic data mining tools can address complex issues such as decision, problem, or task of classification, estimation, association clustering, prediction of a specific task that a human often inherently perform. Here the goal is to make an explanatory decision based on visualization and prediction. Visual data mining tool is the main focus to initiate this process. This will also be explained in detail through multiple case studies in our next book.

Predictive analysis needs to relate the context with respect to time, space and nature of the events or incidents. This includes finding out the answers of when, where and what questions relating to the context based incidents.

The objective of the IRS' VST feature is to help the humans to find out the following context based information for the situation awareness and assisting in making decision:

- When happened?
- Where that happened?
- What is happening now?
- What is going to happen?
- Take action based on the above fours

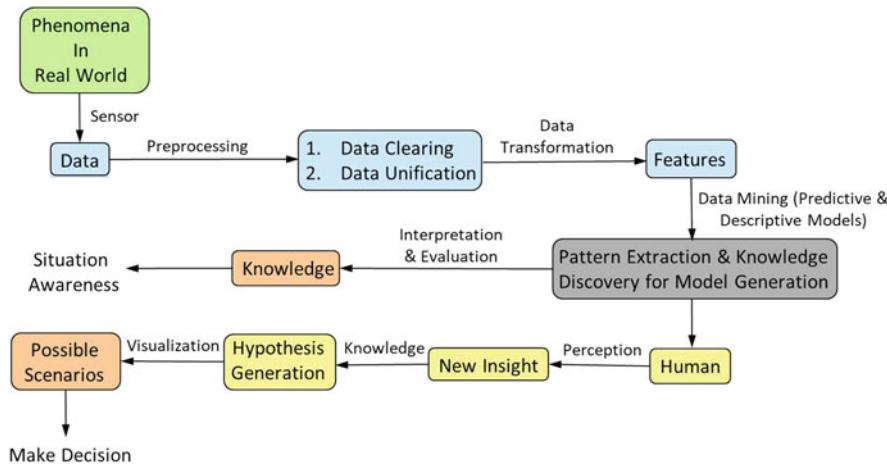


Fig. 31.10 Visualization and prediction using visual story telling

31.6 Virtual Environment for Personal Assistance

Virtual Environment (VE) is a field of study that aims to create a system that provides a synthetic experience for its users. The experience is synthetic, illusory, or virtual: the sensory stimulation to the user is simulated and generated by the system.

Extended Reality (XR) is a place holder for emerging technology. This is integrated with computer generated environments to merge the physical and virtual worlds and gives users an Immersive Virtual Environment (IVE). The VE synthesizes the sensual information into perceptions of environments where the contents excite and stimulate users' perception and interaction. A variety of hardware and software systems are used for the VE related integrated development interface (IDE). This includes displaying the virtual environment, tracking systems, recording users' movements and selecting appropriate portions of the virtual environments to be displayed within the interface.

Virtual Environment (VE) is field of study that aims to create a system that provides a synthetic experience for its users. The experience is synthetic, illusory, or virtual: the sensory stimulation to the user is simulated and generated by the system. In practice, the VE system usually consists of various types of displays for stimulation, sensor to detect user's actions, and a computer that processes the user's action and display the output. Figure 31.11 in reference Kim (2005) is an abstract system view of the VE system.

One important component of a successful VR system is the provision of interaction that allows the user not just to feel a certain sensation, but also to change and affect the virtual world in some way.

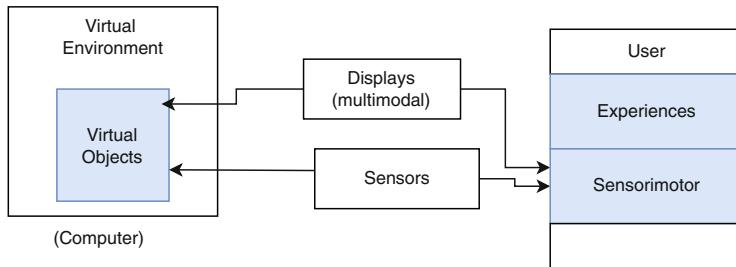


Fig. 31.11 Abstract view of a VE system (Kim 2005)

31.7 Sensor Fusion

Understanding how human perceives and transmits various signals through different modality channels is important for designing a multi-modal system. In addition, the interpretation of multi-modal signals in designing a multi-modal system with an appropriate level of fusion and fission of the information in the modality channels is important. The information in different input and output modality channels can be different.

The fusion of information refers to the analysis and integration of input information that arrives through different modalities into a composite input, i.e., to a meaningful communicative act, whereas the fission of information of information is the opposite process at the generation side and refers to the division of information into appropriate modalities so as to generate an efficient presentation identifying three levels of fusion such as lexical, syntactic, and semantic. Lexical fusion happens on the hardware and software levels, for example, when selecting objects with the shift key down. The syntactic fusion involves combining data to form a syntactically complete command, whereas the semantic fusion concerns the detailed functionality of the interface and defines the meanings of the actions.

Human–Robot Interaction usually requires the robot initiate to approach a specific person or to wait until some people come for help. As a consequence, the robot needs to understand the human's behavioral intention and to decide whether interaction with a specific person should be launched or not. In sensor fusion, available information are organized to make the information accessible for making decisions.

Multi Sensor Data Fusion

Data fusion is the process by which information from multiple sensors and sources can automatically be filtered, aggregated and extracted, data integrated and interpreted to maximize useful information content, improve reliability and discriminatory performance, minimizing the amount of data retained in the ultimate by learning to fuse data from multiple sensors. A multi-sensor data fusion aims to combine multiple sensor data to produce conclusions, which cannot be achieved with only one sensor.

Multimodal sensor fusion in intelligent interactive machine processes and intermediates different sense of information; thus the co-ordination and sensing information take place in order as needed.

31.8 Intelligent Human Machine for Communication: Application Scenario

Human interactions are multi-modal. Each interaction uses multiple modes for instance to listen, perceive, sense, gesture, visualize, taste and such interactions are distinguished by modes, modalities and mediums. The modes are related to human sensory systems such as visual, auditory, and tactile. The modality can be perceived as text, images, and tactile sensation that are not represented internally by the machine. The medium is an output device such as a screen, speaker or haptic technology i.e. a feedback device. The medium can be an interactive modular system composed of independent elements.

Multi-modal communication is a combination of multiple heterogeneous sources for interactions. Effective communication is a meaningful representation of multiple data sources. Multi-modal systems process information from different human communication channels at multiple levels of abstraction. These systems emphasize abstract levels of processing, explicit representations of the dialogue context, the user, and investigations of the user's beliefs, intentions, attitudes, capabilities, and preferences. These components are media, mode analysis and design, interaction and context management, user modeling and knowledge sources. Figure 31.12 shows how an intelligent machine is used to interact with the human using different modalities such as speech, gestures, images, graphics and use virtual personal assistance. The detailed will be captured in the upcoming text book.

31.9 Background Information

General

'Communication' is a Latin word. It literally means something in common, or a relationship with someone in common. Human can easily recognize an object after visualizing this. To automate a machine to recognize It has stated in 1922, first a word based , then 1962 which is used to recognized 16 words, 1972 was 100 words connected speech recognition, 1980s connected speech recognition using HMM, then 10k+w using machine learning, then deep neural network is used in 2010. 1 million plus word in 2016 using deep neural network system.

The Spoken dialogue management (SDM) is responsible for controlling the content and the flow of the spoken dialogue, it provides a key factor in ensuring a user-friendly and consistent interaction.

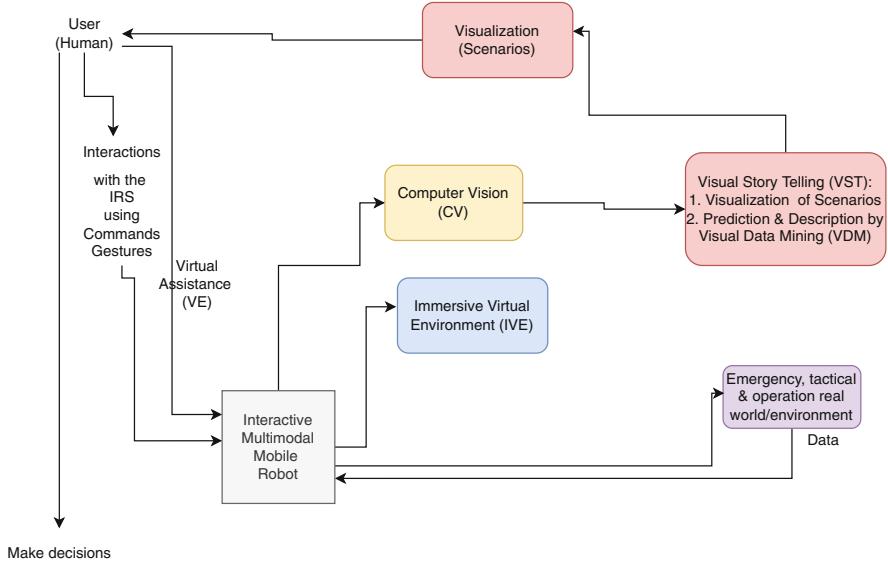


Fig. 31.12 Application Scenario of intelligent interactive machine

Since an SDM is responsible for controlling the content and the flow of a spoken dialogue, it provides a key factor in ensuring a user-friendly and consistent user-system interaction. An object is challenging. A deterministic computer reasons by a clear if-then-else, rule-based structure. This means, the system has an output for each input. The system output will always be the same except if something fails.

The VR applications are widely used in training high risk taking people to perceive possible dangerous, expensive, impossible and any such rare situations. In order to gather desired experience using the VR, the undertaking environment needs to be carefully chosen and designed. The concept of VR was initially formulated in the 1960s.

An autonomous system probabilistically given a set of inputs, meaning that it makes guesses about best possible courses of action given sensor data input. An intelligent machine should:

- perceive its environment,
- affect its environment,
- reason about observations and actions,
- learn from observations and actions,
- have goals.

A wide range of subject is involved in this interdisciplinary multimodal human machine interaction. This will be covered in details in our upcoming book.

The topic will be discussed in details in “Multimodal Interactive Robot in Communication” text book. References used in this chapter among others are

Chaudhary (2018), Bulzacki (2015), Saxena (2020), Soukup (2002), Shapiro and Stockman (2001), Kim et al. (2018), and Brahnam and Jain (2011). Readers are encouraged to read these.

31.10 Exercises

Exercise 1 Prepare pilot study project for a nursing home that can make use of multimodal interactive robot as a companion.

Exercise 2 Prepare a pilot study for a multimodal healthcare gesture recognition interface based communication system between patients and doctors through gestures

References

- [Chaudhary2018] Chaudhary, A., Robust Hand Gesture Recognition for Robotic Hand Control, Springer, 2018
- [Bulzacki2015] Bulzacki,A., Machine Recognition of Human Gestures Through Principal Joint Variable Analysis, PhD thesis, Ryerson University, 2015
- [Saxena2020] Saxena, P., <https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/>, 2020
- [Soukup2002] Soukup, T., Davidson, I., Visual Data Mining: Techniques and Tools for Data Visualization and Mining, John Wiley & Son, 2002
- [Shapiro2001] Shapiro, L. G., Stockman, G., Computer Vision. Prentice-Hall, 2001
- [Kim2005] Kim, G. J., Designing Virtual Reality Systems The Structured Approach, Springer-Verlag, London, 2005.
- [Jain1995] Jain, B. G. S. R., Kasturi, R., Machine Vision, 2nd ed. McGraw-Hill Education, August 1995.
- [Kimetal2018] Kim, S. J., Jeong, Y., Park, S., Ryu, K., Oh, G., Augmented Reality and Virtual Reality, T. Jung and M. C. T. Dieck, Eds. Manchester, UK: Springer, 2018.
- [Brahnam2011] Brahnam, S., Jain, L. C., Advanced computational intelligence paradigms in healthcare 6: virtual reality in psychotherapy, rehabilitation, and assessment. Springer-Verlag Berlin Heidelberg, 2011.

Chapter 32

Comparisons



Overview

There are many different methods, concepts, and implementation issues concerning digital signals. We discussed throughout this book what they have in common. However, there are many aspects and details not answered. This is not surprising when one looks at the broadness of the topic. The applications have in common that they originate from the outer world including, for instance, other humans and they produce results that are understandable by humans. So far, we have considered methods from the perspective of different topics. Despite the fact that these methods are manifold they have common aspects for each topic. In this chapter, we now look at the main differences of for given topics. This brings additional clarification to the motivations of the approaches.

32.1 Generalities

To properly implement a design in specific DSP hardware familiarity with the specifics of development tools that relate to it is crucial. For a given process, the question is what can we learn about it? Despite the fact that such processes are very diverse, one can state that there are common principles underlying them. Information about the unknown elements is necessary. One needs methods to capture them and here lie the major challenges. To address these challenges, machine learning techniques include supervised and unsupervised learning are introduced here. Instances of these techniques are covered under clustering. The development of practical applications is guided by principles throughout the chapters where the problems and differences are exhibited in their realizations. A hardware point of view is also addressed, discussing digital signal processors. More recently, microcontrollers are a specialized microprocessor/Digital Signal Processors (DSP) with

its architecture optimized for the operational needs of digital signal processing. However, this is not the main topic of the book.

Briefly we will discuss the advantages of using the fixed- and floating-point formats, and micro-controllers (Arduino's and Raspberry Pi's) in developing signal processing applications. This can be illustrated by contrasting the requirements of two common video and audio processing applications. Video has a high sampling rate that can amount to tens or even hundreds of megabits per second (Mbps) in pixel data, depending on the application. Pixel data is usually represented in 1 to 4 streams representing independent planes of the image. For each pixel the corresponding data is encoded with integer data types, e.g., one for the red, green, and blue (RGB). In most systems, each color requires 8 to 12 bits, though advanced applications may use up to 14 bits per color. Key mathematical operations of the industry-standard MPEG video compression algorithms include discrete cosine transforms (DCTs) and quantization.

Audio, by contrast, has a more limited data flow of about 1 Mbps that results from 24 bits sampled at 48 kilosamples per second (ksps). A higher sampling rate of 192 ksps will quadruple this data flow rate in the future, yet it is still significantly less demanding than with video. Operations on audio data include infinite impulse response (IIR) filtering.

For this reason, the stochastic processes from a general point of view is first discussed. As for practical applications, we will consider them through the lens of historical developments by considering different aspects provided in the chapters. Video signal processing thus has much more raw data to process than audio. DCTs and quantization are handled effectively using integer operations, which together with the short data words make video a natural application for fixed-point DSPs. The massive parallelism of the some of the DSPs makes them an excellent platform for applications that run multiple video channels.

Video may have a larger data flow, but audio has to process its data more accurately, hence precision is more important for audio applications than for video applications. Although audio has usually been implemented using fixed-point devices, high-fidelity audio today is transitioning to the greater accuracy of the floating-point format. This necessity to increase precision of sample representation requires wider words DSPs (24-bit signal, 24-bit coefficients, 53-bit intermediate product) to provide much greater accuracy in audio output, resulting in higher sound quality.

Sampling sound with 24 bits of accuracy yields 144 dB of dynamic range, which provides more than adequate coverage for the full amplitude range needed in sound reproduction. Wide coefficients and intermediate products provide a high degree of accuracy for internal operations, a feature that audio requires for at least two reasons.

The floating-point format by its nature aligns well with the sensitivity of the human ear and becomes more accurate as floating point numbers approach 0. This is the result of the exponent's keeping track of the significant zeros after the binary point and before the significant data in the mantissa. This is in contrast to a fixed point system for very small fractional numbers. All of these aspects of floating-point real arithmetic are essential to the accurate reproduction of audio signals.

The other types of applications also lend themselves better to either fixed or floating-point computations. Today, one of the heaviest uses of DSPs is in wired and wireless communications, where most data is transmitted serially in octets that are then expanded internally for 16-bit processing based on integer operations. Obviously, this data set is extremely well-suited for the fixed-point format, and the enormous demand for DSPs in communications has driven much of fixed-point product development and manufacturing. Floating-point applications are those that require greater computational accuracy and flexibility than what fixed-point DSPs offer. For example, image recognition used in medicine is similar to audio in requiring a high degree of accuracy. Many levels of signal input from light, x-rays, ultrasound and other sources must be defined and processed to create output images that provide useful diagnostic information. The greater precision of signal data, together with the device's more accurate internal representations of that data, enable imaging systems to achieve a much higher level of recognition and definition for the user.

Contrary to imaging applications requiring fixed-point representation, in radars for navigation and guidance traditionally floating-point is used since it requires a wide dynamic range that cannot be defined ahead of time. The radar system may be tracking in a range from 0 to infinity, but needs to use only a small subset of the range for target acquisition and identification. Since the subset must be determined in real time during system operation, it would be all but impossible to base the design on a fixed-point DSP with its narrow dynamic range and quantization effects.

Wide dynamic range also plays a part in robotic design. Normally, a robot functions within a limited range of motion that might well fit within a fixed-point DSP's dynamic range. However, unpredictable events can occur, for instance, the robot might weld itself to an assembly unit, or something might unexpectedly block its range of motion. In these cases, feedback is well out of the ordinary operating range, and a system based on a fixed-point DSP might not offer programmers an effective means of dealing with the unusual conditions. The wide dynamic range of a floating-point DSP, however, enables the robot control circuitry to deal with unpredictable circumstances in a predictable manner.

The critical feature for designers of algorithms is the greater mathematical flexibility and accuracy of the floating-point formats, because they offer them flexibility in applying arithmetic with greater precision and a wider dynamic range. That is the reason for rapid raise of micro-controllers (e.g., Arduino's and Raspberry Pi's); providing flexibility to deal with wide range of signals (Analog and Digital) and peripheral devices (from monitors, keyboards, motors, cameras, to ultrasound and other sensors) coupled with processing power. Processor chip's flexibility (8-bit, 16-bit and 32-bit), coupled with online instructions (<https://www.arduino.cc/>, <https://www.raspberrypi.org/>), and online examples and learning plans (<https://www.arduino.cc/education>, <https://www.raspberrypi.org/learn/>) provided an adequate mix of capabilities for using the named devices for wide range of applications.

As of this writing, there are a huge number of hardware-software platforms that determine what kind of applications can run. For example, we are writing this

section of the book on a laptop computer, but we are also using other personal laptops or desktop computers to access the same content over the web based tool called <https://www.overleaf.com/>. The first computers were programmed with machine instructions, while today users program their computing platforms with scripting language called “python”. While there are limitation imposed with each computing platform (desktop, laptop, tablet, “smart” phone—apple or android, etc.) it is hard to imagine hitting those limitations.

Computer platform typically is defined by the tool-set that it uses. This includes application and user interfaces that are used to configure, customize, design and develop software services. The following is a common categorization of computing platforms:

- **Operating system**—provides software that provides for development, management and execution of variety application software. It helps to manage a computer’s hardware resources as well as supporting basic functions like scheduling tasks, and controlling peripherals. The most popular operating systems (OS) are listed below
 - Windows
 - Mac OS
 - Linux
- **Hardware Platforms**—the OS’s are run in variety of platforms listed below:
 - Client/Server
 - Mobile Platform
 - Cloud Platform
 - Arduino, Raspberry Pi, Digital Signal Processor (DSP), other

Most DSP’s, micro-controllers and general processors use two’s complement fractional number representations. Those representations typically are utilizing in different so-called Q formats. The native formats for the DSP family are a signed fractional $Q1.(N-1)$ and unsigned fractional $Q0.N$ format, where N is the number of bits in the data word. The computer hardware (DSP or more general processor) will dictate what kind of representation the processor supports. A hardware platform refers to computer’s processor architecture. For example, the x86 and x86-64 CPUs constitute the most common computer architectures.

A summary of all chapters, examples, and comparisons is captured here:

32.1.1 EEG and ECG

Both are applications from the medical domain. However, they are of different kinds and need different methods for feature extractions. The difference starts with the sources. The signal processes are not the same and therefore the methods for handling them too.

32.1.2 Speech and Biomedical Applications

In speech as well as in biomedical applications the signals come both from the human body. They both contain some information. The difference is that in speech the sender has some intention. The biomedical signals reflect properties of the body that contain information with respect to possible diseases and dangers to health. This information cannot easily be understood by humans. Pathological aspects result in pain or uncomfortable feeling. In some sense, the signals in speech are of a more difficult nature. They come in the first place in the form of wave and one has to transform them into other signals. Understanding of the signals can lead to a possible treatment of the underlying reason. The detailed analysis of biomedical signals tells however that the understanding of their meaning is equally difficult. The main technical differences between speech and biomedical treatments are in the following.

- (a) The semantics
- (b) Data preparation
- (c) Recognition.

Discussion:

- (a) The semantics in speech is related to the uttered words. In order to understand them properly one needs knowledge about the topic. There is no organ like the ear to understand biomedical signals. In order to understand them one needs the brain to analyze the measurements.
- (b) The data preparation for speech quite different.
- (c) The recognition of the data themselves is easier for biomedical signals. However, this is different for the properties in particular because the properties coming from medicine are partially unknown.

32.1.3 Seismic and Biomedical Signals

Seismic signals have their origin in the earth. They contain information about the earth that can be used for predictions. Quite important are the predictions about possible earthquakes. The prediction is not used for treatment but rather for an adequate behavior. Biomedical signals share many aspects of Seismic signals and processes.

32.1.4 Speech and Images

Speech and images are two media with applications of quite different character. They both mainly address direct communication with humans. The difference does

not only come from the reception of humans. From the machine point of view images have a standard representation in a quadratic form while speech does not. There are several structural and computational differences between speech and images that we will discuss now.

- (1) For speech one has a specific sender and receiver but not for images. Speech has a temporal aspect and images do not. These temporal properties are an essential aspect of psychoacoustic that influences the meaning of the speech. In images psychoacoustic or an equivalent does not play a big role and therefore temporal aspects are irrelevant for a single image. For images the temporal aspect can be added in order to obtain videos and movies. For this reason one has developed movies and videos that have a temporal ordering. However, each movie presents a set of images. We did not discuss movies in the book.
- (2) In speech one gets the signals step by step. This sequence contains several parameters as loudness of signals and distances between signals. In an image one gets all pixels at the same time and there are many pixels. The pixels are not presented in a linear or related ordering but rather in a square or in three dimensional ways. Instead of a “sooner or later relation” there is a distance measure between the pixels that are important for the interpretation of the image.
- (3) Pixels have a direct visual impression to humans while speech has it only on the level of words. The visual impression requires for further processing a description on an abstract level. This needs to go from the elementary signal level to a higher level.
- (4) In images the totality of the pixels contains all the information of the message. In speech we need to extract features. This is parallel to combine pixels to certain sets that are useful to the interpretation. This can be done in different ways which lead to different results.
- (5) In many cases speeches result directly in text. This is not the case for images. For images there are many possibilities to describe them with text.
- (6) In speech psychoacoustic phenomena play an important role. If they occur in images they are of a completely different nature.

32.2 Overall

These differences have not only a crucial impact on the creation and formal representation of speech and images but also on the recognition. For images geometric aspects play a role while for others like speech temporal aspects play a role. Humans have the ability to grasp the intention from images and music in many situations. Psychoacoustic properties have been developed for some signal types. They play a major role in medical applications. Also, in other applications they often play a crucial role and they are invented quite often from scratch. Such comparisons are important for institutions where signals arrive from different sources.

32.3 Background Information

The first computers were programmed with machine instructions, while today users program their computing platforms with scripting languages like “python”. While there are limitation imposed with each computing platform (desktop, laptop, tablet, “smart” phone—apple or android, etc.) it is hard to imagine hitting those limitations.

32.3.1 *General*

The comparison of signals takes usually place on the level of basic signals, mainly from the computationally view. The comparison of signal types from the view of applications has not been treated very much. The comparison is in the first place of scientific interest. On the other hand it is of practical use if one has several types of signal processes to deal with.

The signals themselves are too large for combinatorial treatment. To shorten the length, one uses features. There are several ways to extract them. They depend on the information one is interested in and therefore there are many possibilities for this purpose. Here we compare such extraction methods.

For each intention, one has to choose a specific form of signals. These forms depend mainly on the purpose of the application that determines the source of the signals and the source describes to a large degree the kind of the signal processes. The choice of the extraction method is crucial and the book spends much for this. Such methods are different for each signal type. Here we compared the individual type methods. Such comparisons are useful for institutions where different kinds of sign processes play a role.

32.4 Exercises

Exercise 1 Give an example of some speech describing an image and compare it with the image representation.

Exercise 2 Work on an image with the methods from speech and discuss the result.

Glossary

Auto-Regressive Moving Average Auto-Regressive Moving Average is a kind of process. [48](#)

Automatic Speech Recognition The Automatic Speech Recognition is an interdisciplinary sub field of the computer science and computational linguistics that develops methodologies and technologies to enable the recognition and translation of spoken language into text by computers (Wikipedia). [349, 350](#)

Best Linear Unbiased Estimator Best Linear Unbiased Estimator is an estimator based on the ordinary least squares estimator which is linear, minimum, and unbiased. [123](#)

CCITT Consultative Committee for International Telephony and Telegraphy. [24](#)

Connected Word Speech Recognition connected word speech recognition where each utterance consists of a speech of concatenated word sequence. [347](#)

Digital Signal Processor Digital Signal Processor is a type of microprocessor focusing on operations related to signals. [24](#)

Dynamic Time Warping DTW is an algorithm for computing a similarity between two sequences. [179, 211](#)

Electrocardiogram Electrocardiogram is a rendering of a biological signal. [42](#)

Electromyography Electromyography is a rendering of a biological signal. [42](#)

Expectation Maximization Expectation Maximization is an optimization algorithm for training parameters such that a model fits encountered data. [257](#)

Finite Impulse Response digital filter Finite Impulse Response digital filter. [40](#), [98](#)

Gaussian Mixture Model GMM is often categorized as a clustering algorithm, fundamentally it is an algorithm for density estimation. It is a parametric probability density function represented as a weighted sum of Gaussian component densities. [258](#), [262](#)

Hidden Stochastic Model HSMs are used for modelling general stochastic processes with some un-observable random variables. [196](#)

Human–Robot Interaction Human–Robot Interaction is the study of interactions between humans and robots. It is often referred as HRI by researchers. Human–robot interaction is a multidisciplinary field with contributions from human–computer interaction, artificial intelligence, robotics, natural language understanding, design, and psychology (Wikipedia). [592](#)

Independent Component Analysis Independent Component Analysis subdivides a signal into separate independent sources linearly mixed in several samples. [162](#)

Infinite Impulse Response digital filter The Infinite Impulse Response digital filter. [40](#), [98](#)

Isolated Speech Recognition isolated speech recognition where each utterance consists of simple word or phrase. [347](#)

Karhunen–Loeve Transform Karhunen–Loeve Transform is a kind of transform. [83](#)

Linear Minimal Mean Square Error Estimator Linear Minimal Mean Square Error Estimator is an estimator based on the ordinary least squares. [128](#)

Linear Predictive Coding Coefficients LPC are a family of features that can be extracted from sound signal. [231](#)

Local Cosine Transform A transform called Local Cosine Transform. [69](#)

Maximum A Posteriori Maximum A Posteriori is an approach to optimization maximizing the posterior probability of a hypothesis given the observations. [120](#)

Maximum Likelihood Maximum Likelihood is an optimisation approach maximizing the likelihood of the available observations. [120](#)

Mel-Filter Cepstrum Coefficients MFCC are a family of features that can be extracted from sound signal based on a filtered FFT spectrum. [221](#)

Minimal Mean Square Error Estimator Minimal Mean Square Error Estimator is an estimator based on the ordinary least squares. [126](#)

Monte Carlo Monte Carlo is a type of sampling where each variable is estimated based on its probability. [149](#)

Perceptual Linear Prediction Cepstrum Coefficients Perceptual Linear Prediction Cepstrum Coefficients. [221](#)

Principal Component Analysis Principal Component Analysis subdivides a signal into separate independent sources linearly mixed in several samples when the data is Gaussian, linear, and stationary. It finds the directions that maximize the variance in the dataset. [164](#)

Quadratic Mirror Filterbank Quadratic Mirror Filterbank subdivides a signal into octave bands using critical sampling. [148](#)

Short Time Fourier Transform A transform called Short Time Fourier Transform. [40, 69](#)

Signal to Noise Ratio The Signal to Noise Ratio is defined as the ratio of signal power to the noise power, often expressed in decibels. [19, 315](#)

Single Input and Single Output System Single Input and Single Output System. [50](#)

Unweighted Least Square Unweighted Least Square. [271](#)

VA Visual Analytics. [531](#)

Vector Quantization Vector Quantization is a method of classifying features. [256](#)

Visual Story Telling Visual Story Telling is to Observe the world, collect the data, and interpret them using visual elements and symbols so that the sensible and meaningful stories can be revealed and predicted. The objective of such a transformation typically infers finding, judging or observing for making initial decisions. [531](#)

Wake Up Word Wake-up-word is a special speech recognition technology capable of detecting it when spoken in alerting context. [476](#)