Tokunbo Ogunfunmi · Roberto Togneri
Madihally (Sim) Narasimha *Editors*

Speech and Audio Processing for Coding, Enhancement and Recognition

Springer

Speech and Audio Processing for Coding, Enhancement and Recognition

Tokunbo Ogunfunmi • Roberto Togneri
Madihally (Sim) Narasimha
Editors

Speech and Audio Processing for Coding, Enhancement and Recognition

Springer

Editors
Tokunbo Ogunfunmi
Department of Electrical Engineering
Santa Clara University
Santa Clara, CA, USA

Roberto Togneri
School of EE&C Engineering
The University of Western Australia
Crawley, WA, Australia

Madihally (Sim) Narasimha
Qualcomm Inc.
Santa Clara, CA, USA

ISBN 978-1-4939-1455-5 ISBN 978-1-4939-1456-2 (eBook)
DOI 10.1007/978-1-4939-1456-2
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2014951041

Preface

It is without doubt that we live in an interconnected world where we are always within reach of smartphone, tablet or telephone system and through which we are always communicating with friends and family, colleagues and workmates or an automated voicemail or interactive dialog system. Otherwise we just relax and switch on the radio, stream some music or watch a movie. These activities are part of our everyday lives. They have been made possible through the advances in speech and audio processing and recognition technologies which only in the last decade have seen an explosion in usage through a bewildering array of devices and their capabilities.

Speech coding refers to the digital representation of the information-bearing analog speech signal, with emphasis on removing the inherent redundancies. Efficient coding of speech waveforms is essential in a variety of transmission and storage applications such as traditional telephony, wireless communications (e.g., mobile phones), internet telephony, voice-over-internet protocol (VoIP) and voice mail. Many of these applications are currently going through an impressive growth phase.

Speech recognition encompasses a range of diverse technologies from engineering, signal processing, mathematical statistical modelling and computer science language processing necessary to achieve the goal of human–computer interaction using our most natural form of communication: speech. Applications of speech recognition have exploded due to the advent of smartphone technology where the use of the traditional keyboard and mouse has given way to touch and speech and in enterprise automated computer voice response services for enquiries and transactions. We are now experiencing an exponential growth in the adoption of speech recognition in smartphone and mobile technology, in information and transaction services and increased R&D effort on efficient low-cost and low-power implementations, robustness in the presence of ambient noise and reliable language understanding and dialog management.

In this book we provide readers with an overview of the basic principles and latest advances across a wide variety of speech and audio areas and technologies across ten chapters. These are organized into three parts from front end signal processing involved with speech coding and transmission and the more sophisticated approaches deployed for speech enhancement to the back end user interface involved with speech recognition to the latest "hot" research areas in emotion recognition and speaker diarization. This book brings together internationally recognized researchers across these diverse fields spanning many countries including the USA, Australia, Singapore and Japan from leading research universities, industry experts from Microsoft and Qualcomm and front line research institutions like Microsoft Research, USA, Institute for Infocomm Research, Singapore and NTT Labs, Japan.

We have divided the book into three parts: "Overview of Speech and Audio Coding", "Review and Challenges in Speech, Speaker and Emotion Recognition" and "Current Trends in Speech Enhancement".

Part I comprises four chapters.

The first chapter traces a historical account of speech coding from a front-row seat participant and is titled "From 'Harmonic Telegraph' to Cellular Phones". The second chapter gives an introduction to speech and audio coding, emerging topics and some challenges in speech coding research. In the third chapter, we present scalable and multirate speech coding for Voice-over-Internet Protocols (VoIP) networks. We also discuss packet-loss robust speech coding. The fourth chapter details the recent speech coding standards and technologies. Recent developments in conversational speech coding technologies, important new algorithmic advances, and recent standardization activities in ITU-T, 3GPP, 3GPP2, MPEG and IETF that offer a significantly improved user experience during voice calls on existing and future communication systems are presented. The Enhanced Voice Services (EVS) project in 3GPP that is developing the next generation speech coder in 3GPP is also presented.

Part II includes four chapters which cover the depth and breadth of speech and audio interfacing technologies. The part starts with two overview chapters presenting the latest advances and thoughts in statistical estimation and machine learning approaches to feature modelling for speech recognition, specifically ensemble learning approaches and dynamic and deep neural networks. This is followed by two chapters representing new and emerging research and technology areas which extend speaker recognition to: how speech can be used to detect and recognize the emotional state of a speaker instead, to the deployment in the real world task of speaker diarization in room conversations, that is who spoke when.

Part III presents two different alternative paradigms to the task of speech enhancement. Assuming the availability of multiple microphone arrays the first chapter in this part deals with speech enhancement in the widest sense where speech is degraded by interfering speakers, ambient noise and reverberations and provides a framework which integrates both spatial and spectral features for a blind source separation and speech enhancement solution. The second and final chapter in this part presents a more fundamental approach for signal channel speech enhancement in the presence of ambient additive noise based on the modulation

spectrum approach for differentiating and separating time-frequency speech features from the additive interfering noise features.

The convergence of technologies as exemplified by smartphone devices is a key driver of speech and audio processing. From the initial speech coding and transmission to the enhancement of the speech and the final recognition and modelling of the speech we have in our hands that smart phone device that can capture, transmit, store, enhance and recognize what we want to say. This book provides a unique collection of timely works representing the range of these processing technologies and the underlying research and should provide an invaluable reference and a source of inspiration for both researchers and developers working in this exciting area.

We thank all the chapter contributors which includes Bishnu Atal, Jerry Gibson, Koji Seto, Daniel Snider, Imre Varga, Venkatesh Krishnan, Vivek Rajendran, Stephane Villette, Yunxin Zhao, Jian Xue, Xin Chen, Li Deng, Vidhyasaharan Sethu, Julien Epps, Eliathamby Ambikairajah, Trung Hieu Nguyen, Eng Siong Chng, Haizhou Li, Yasuaki Iwata, Tomohiro Nakatani, Takuya Yoshioka, Masakiyo Fujimoto, Hirofumi Saito, Kuldip Paliwal and Belinda Schwerin for their work.

We thank Springer Publishers for their professionalism and for support in the process of publishing the book. We especially thank Chuck Glasser and Jessica Lauffer.

We hope the material presented here will educate new comers to the field and also help elucidate to practicing engineers and researchers the important principles of speech/audio coding, speech recognition and speech enhancement with applications in many devices and applications such as wireless communications (e.g., mobile phones), voice-over-IP, internet telephony, video comm., text-to-speech, etc. which are ubiquitous today.

Santa Clara, CA, USA — Tokunbo Ogunfunmi
Crawley, WA, Australia — Roberto Togneri
Santa Clara, CA, USA — Madihally (Sim) Narasimha
June 2014

Contents

Part I
Overview of Speech and Audio Coding

Chapter 1
From "Harmonic Telegraph" to Cellular Phones

Bishnu S. Atal

> Leave the beaten track occasionally and dive into the woods. Every time you do so you will be certain to find something that you have never seen before. Follow it up, explore all around it, and before you know it, you will have something worth thinking about to occupy your mind. All really big discoveries are the results of thought.
>
> (Alexander Graham Bell)

Abstract It all started with two patents issued to Alexander Graham Bell in March 1876 and the world changed forever. Vast distances began to shrink. Soon, nobody was isolated. The invention produced a new industrial giant whose research laboratories supported the best in scientific research and engineering leading to major technical advances of the twentieth century. The desire for communication, anytime, anywhere spread fast; stationary phones connected by wires started fading away, replaced by mobile phones or "cellular phones" reflecting the cell structure of the wireless medium. The book chapter will provide a history of the telephones, starting from Alexander Graham Bell's "harmonic telegraph" in 1876 to modern cellular phones.

1.1 Introduction

In the middle of the nineteenth century, before the invention of the telephone, the telegraph was the primary form of long distance communication. The telegraph was the fastest and reliable way to transmit information, but was limited to sending or receiving one message at a time. The telegraph message traffic was rapidly expanding and Western Union, which ran the telegraph business, was trying to find a way to send multiple messages on each telegraph line to avoid the cost of constructing new lines. Copper for wires was a major expense then for the telegraph company and sending multiple messages on a single telegraph line was an immediate and pressing need.

B.S. Atal (✉)
University of Washington, Seattle, WA, USA
e-mail: bsatal@bishnu.net

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_1

1.1.1 The Multiple Telegraph "Harmonic Telegraph"

Because the need was so obvious, a number of inventors were busy. Thomas Edison began designing devices for multiple (duplex and quadruplex) telegraphy in 1865 and he had the technical background due to his experience as a telegraph operator [1]. Due to his training in electrical signals, Edison was trying to solve the problem by manipulating the electrical current.

In 1873, Alexander Graham Bell, who was professor of Vocal Physiology at the Boston University, and taught deaf students to speak, began experimenting with a device, which could send several telegraph signals simultaneously over a single wire. He knew that others had transmitted musical tones over a wire by using the intermittent (dot & dashes) current of telegraphy. He thought of sending more than one tone over the same wire simultaneously and then separate the tones at the receiving end. He called his version of the multiple telegraph the "harmonic telegraph".

With many years of scientific training, Bell had gained an extraordinary understanding of the way sounds of speech are created and heard. He carried out a series of experiments to determine how different vowel sounds are produced. He concluded that every vowel sound is a combination of resonances from different cavities of the mouth.

For several years Bell continued to develop a functioning harmonic telegraph. He used to do his experimenting on harmonic telegraph at night, as he was busy during the day at the Boston University. Although Bell was a teacher by profession, he was thinking seriously about the commercial side of his work. In 1875, Bell together with Gardiner Hubbard (a lawyer), and George Sanders (businessman) established the "Bell Patent Association". Bell hired Thomas Watson to assist him in his research. Watson was an experienced machinist and would help towards the development of the harmonic telegraph. Both Bell and Watson lived in two cheap little bedrooms. Watson's wages of nine dollars a week were being paid by Sanders and Hubbard.

1.1.2 Bell's Theory of Transmitting Speech

Bell explored the possibility of "telegraphing" speech, though he then had no idea how to go about doing it. One evening Bell said to Thomas Watson: "Watson, I want to tell you of another idea 1 have, which I think will surprise you. If I could make a current of electricity vary in intensity, precisely as the air varies in density during the production of sound, I should be able to transmit speech telegraphically." But Bell's partners, Hubbard and Sanders, were insisting that the wisest thing for Bell to do was to perfect the harmonic telegraph then he would have money to build air castles like "telephone".

1.2 Early History of the Telephone

1.2.1 The Telephone Is Born

It was through his experiments with the harmonic telegraph, plus his knowledge of music and human speech and hearing, that Bell found the way to the telephone. Bell and Watson continued their work. A great breakthrough came in 1875. It was actually an "accident". While testing the harmonic telegraph device between two rooms in the electrical shop, Bell heard a faint but distinct sound. There are a number of accounts as to exactly what happened on that memorable day of June 2, 1875, but Watson's own words tell the story dramatically [2]. Bell's great success marked not only the birth of the telephone but the death of the harmonic telegraph as well.

Alexander Graham Bell filed several patent applications for his work on harmonic telegraph; the most important was the US patent 174,465, filed February 14, 1876 and issued on March 7. The patent was titled "Improvements in Telegraphy" and described new and useful improvements in telegraphy for transmitting vocal or other sounds telegraphically by causing electrical undulations, similar in form to the vibrations of the air accompanying the vocal or other sounds. The key point to Bell's application, the principle of variable resistance, was scribbled in a margin on the rough draft, almost as an afterthought. Some 600 lawsuits would eventually challenge the patent.

Bell's telephone "the speaking telegraph" was not universally welcomed. Some people dismissed it as a scientific toy of little value. By the fall of 1876, Bell and Hubbard offered to sell the telephone patent rights to Western Union Telegraph Company for $100,000. Western Union said no. Western Union believed that the telegraph, not the telephone, was the future. Only a few months later, Western Union realized what an unwise decision they had made. The telephone began to make its way into society, catching the public imagination as people preferred two-way conversations over the telephone.

1.2.2 Birth of the Telephone Company

The telephone business was formally organized and on July 9, 1877, The Bell Telephone Company was formed. In December 1877, Western Union created the American Speaking Telephone Company, to conduct its telephone business; Thomas Alva Edison started working for this company. Western Union was a giant then. Bell Telephone had installed only 3,000 phones by then. Western Union, on the other hand, had 250,000 miles of telegraph wire strung over 100,000 miles of route. On November 10, 1879 Bell won its patent infringement suit against Western Union in the United States Supreme Court. The American Telephone and Telegraph Company (AT&T) was established in 1885.

By the end of 1892 there were nearly 240,000 telephones in use in the United States. Bell's patents expired in 1893 and 1894. During the 6 years following the patents' expiration more than 6,000 telephone companies were inaugurated in the United States. AT&T started new efforts based on scientific research and development with emphasis on basic science to fight a battle for its survival.

1.2.2.1 Research at Bell Company

Scientific research at Bell Telephone Company started in 1885 when Hammond V. Hayes joined the company. Hayes, a graduate of Harvard, had studied electrical engineering at MIT in Cambridge, MA and was one of the Harvard's earliest Ph.D in physics. The technical staff at the telephone company consisted of 81 employees on December 31, 1885 and it grew to 195 employees in January 1905, under the leadership of Dr. Hayes. This was the first formal organization of research leading to Bell Telephone Laboratories. Hayes was quick to sense the technical complexity of telephony and to realize that its scientific roots must extend into deeper soil. Early workers involved with the development of telephones had to rely on intuition, ingenuity, and experiment. However, by 1900, a theoretical basis for electrical communication started emerging from the scientific research, both within and outside of the Bell Company.

It is highly unusual for a company to describe its research organization in the annual report to its stockholders. In its 1913 report, the president of AT&T, Theodre N. Vail, made several interesting points. I provide here a summary.

At the beginning of the telephone industry there was no school or university conferring the degree of electrical engineer. AT&T called some of the most distinguished professors of science at many universities to its aid. As problems became more formidable and increased in number and complexity, the engineering and scientific staff was increased in size and in its specialization, so that we now have 550 engineers and scientists. Among them are former professors and instructors of our universities, postgraduate students and other graduates holding various engineering and scientific degrees from 70 different scientific schools and universities, 60 American and 10 foreign institutions of learning being represented. No other telephone company, no government telephone administration in the world, has a staff and scientific equipment such as this. It can be said that this company has created the entire art of telephony and that almost without exception none of the important contributions to the art have been made by any government, telephone administration or by any other telephone company either in this country or abroad.

1.2.2.2 New York to San Francisco Telephone Service in 1915, Nobel Prize, and More

The Bell Company had a policy of hiring the best students of the best professors at the best universities for its research organization. One such person was

Harold D. Arnold, a student of Prof. Robert Millikan of University of Chicago, who recommended him "as one of the ablest man whose research work I have ever directed and had in classes." The telephone network was expanding from New York and Boston towards Chicago and Denver in the west, but with wire thick as rods and with all sort of technological innovations, it was still not possible to hear anything in Denver. Within a few years after his arrival at Bell, Arnold developed a high-vacuum tube amplifier that made it possible to introduce commercial telephone service from New York to San Francisco by 1915.

The talented scientists at Bell continued to advance science and engineering. Another development was the negative feedback amplifier invented by Harold Black in 1927, an invention now regarded as one of the most important discovery. In 1937, Dr. Clinton Davisson became the first Bell Labs person to win the Nobel Prize for his experimental discovery of the wave nature of electrons.

1.3 Speech Bandwidth Compression at AT&T

1.3.1 Early Research on "vocoders"

AT&T's interest in speech bandwidth compression began around 1925 when the company explored the possibility of establishing telephone connection between New York and London. There was a radio–telephone service then operating across Atlantic. The company asked research engineers at Bell Telephone Laboratories if voice signals could be transmitted over existing undersea telegraph cables. The bandwidth required for voice transmission was approximately ten times the bandwidth that was available on the transatlantic telegraph cable. In October 1928, Homer Dudley, an electrical engineer at Bell Laboratories, proposed a device called "vocoder" (voice coder) to compress speech [3]. Ideas behind vocoders remained the central theme of speech coding research for about 35 years [4].

In an excellent review of vocoder research published in 1990, Ben Gold explained Dudley's idea [5]. "To understand Dudley's concept of more telephone channels in the same frequency space, it is important to realize that human speech production depends on relatively slow changes in the vocal-tract articulators such as the tongue and the lips. Thus, if we could develop accurate models of the articulators and estimate the parameters of their motion, we could create an analysis-synthesis system that has a low data rate. Dudley managed to bypass the difficult task of modeling the individual articulators by realizing that he could lump all articulator motion into one time-varying spectral envelope."

Vocoders promised a many-fold reduction of the bandwidth necessary to transmit intelligible speech, but the speech from vocoders was of poor quality unsuitable for commercial use. Why was it so? Manfred Schroeder reviewing the status of speech coding at a talk in 1959 at the Third International Congress of Acoustics in Stuttgart provided the answer [6]. "The answer can be given in two sentences: (1) our ears are

highly discriminating organs, particularly sensitive to the quality of human speech, and (2) no vocoder is capable of reproducing these important quality characteristics in a manner acceptable to the ear". By the time I joined Bell Telephone Laboratories in 1961, the research in speech coding had been discontinued at Bell Labs.

1.3.2 Predictive Coding

In 1965, as part of my Ph.D. course work at Polytechnic Institute of Brooklyn, New York, I took a seminar course on Information Theory, taught by Prof. Mischa Schwartz. We discussed in the course several interesting papers. One of the papers was on predictive coding written by Prof. Peter Elias at MIT. In fact, there were two papers, Predictive coding: I and II, published in the IRE Trans. Information Theory, 1955 [7, 8]. In these papers, Elias mentioned two major contributions that have been made within the past few years. One is Wiener's work on prediction [9] and the other is Shannon's work on the mathematical theory of communication [10]. I provide here a summary of the abstracts of the two papers [7, 8].

> Predictive coding is a procedure for transmitting messages which are sequences of magnitudes. In this coding method, the transmitter and the receiver store past message terms, and from them estimate the value of the next message term. The transmitter transmits, not the message term, but the difference between it and its predicted value. At the receiver this error term is added to the receiver prediction to reproduce the message term. The error terms which are transmitted in predictive coding are treated as if they were statistically independent. If this is indeed the case, or a good approximation, then it is still necessary to show that sequences of message terms which are statistically independent may always be coded efficiently. This is shown in the final section of the second paper.

I found the concept of predictive coding extremely interesting. In discussions with researchers in the speech coding area, I got the impression that such techniques were not applicable for speech coding. In 1966, I started working on my Ph.D. thesis on automatic speaker recognition and I was reluctant to start a side project on speech compression. However, I felt that I should do exploratory investigation to determine if predictive coding could work for speech signals. A first step was to find out if successive samples of prediction error for speech were uncorrelated. I analyzed several segments of voiced speech, band-limited to 3.2 kHz and sampled at 6.67 kHz, using linear prediction with number of predictor coefficients ranging from 2 to 128. The duration of each speech segment was set to 40 ms and linear prediction analysis was carried out using Wiener's formula. The results are illustrated in Fig. 1.1 which shows the spectrum of speech and the prediction error with $p = 16$ and $p = 128$, where p is the number of predictor coefficients. The spectrum of the prediction error at $p = 128$ is nearly white, except at very low frequencies. The results were encouraging, but the important question was whether the prediction error can be encoded at a low bit rate. That is the problem discussed by Peter Elias in his second paper. Manfred Schroeder and I spent next 15 years to find a solution to this problem.

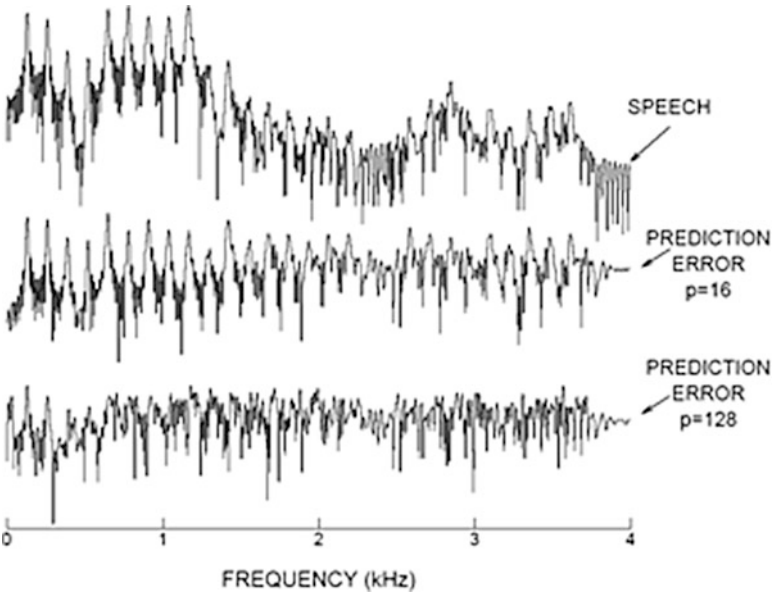

Fig. 1.1 The figure shows spectrum of the speech signal and the corresponding spectra of the prediction error after linear prediction with 16 and 128 coefficients. The speech signal was sampled at 6.67 kHz

1.3.3 Efficient Encoding of Prediction Error

1.3.3.1 Some Comments on the Nature of Prediction Error for Speech

The prediction error is the difference between a speech sample and its predicted value. For linear prediction, the predictor P is a linear filter and the filter $1-P$ is also a linear filter. The spectrum corresponding to the filter $1-P$ is approximately the inverse of the spectrum of the speech signal. The spectrum of the speech signal as a function of frequency varies over a large range, approximately 40 dB (10^4 in power). For voiced speech, the spectrum has large values near the “formants” and near the “harmonics”, but small values in between the formants and the harmonics (see the first plot of Fig. 1.1). Therefore, the spectrum of $1-P$ will be small near the formants and the harmonics, but will be very large in between and will be determined by the low-amplitude portions of the speech spectrum. The prediction error will therefore will have considerable noise in these frequency regions. Since formants and harmonics occur in narrow frequency regions and the regions outside formants and harmonics are large, the prediction error is mostly filled with noise. The situation is analogous to what one finds in inverting a matrix which has a few large eigenvalues but a large number of very small eigenvalues. The resulting matrix after inversion will have a large number of very large eigenvalues which are influenced by very small eigenvalues in the original matrix. The problems created by the ill-conditioned nature of prediction error can be dealt with by developing a proper fidelity criterion.

1.3.3.2 Information Rate of Gaussian Signals with Specified Fidelity Criterion

Shannon considered the problem of quantizing "white" Gaussian signal under a mean-squared error criterion in his original 1948 paper [10] and followed up in another paper [11] published in 1949, where he introduced the concept of a *rate distortion function*. In a paper published in 1964, MacDonald and Shultheiss developed this concept further obtaining results for the case where not only the total mean-squared error but also the spectral properties of the signal was taken into account [12, 13]. The results described in these papers are directly applicable to the prediction error problem.

The rate–distortion viewpoint requires a model for the signal and a fidelity criterion for the error. Let us review briefly some of the results for the rate–distortion function for "white" Gaussian signal with a mean-squared-error distortion measure D [11]. For a Gaussian signal with zero mean and a variance (power) σ^2, the rate distortion function (in bit/sample) is given by

$$R(D) = \max\left[0, \frac{1}{2}\log_2\left(\sigma^2/D\right)\right].$$

The intuitive meaning of this equation is that, for a distortion $D \geq \sigma^2$, we need not send any information ($R = 0$), because we can replace the source output by zeros (and incur an error σ^2 which does not exceed the distortion D). This result can be generalized to the case when the spectrum of either the signal or the noise (or both the signal and the noise) is nonwhite [12]. We will skip the details here. The main idea is to apply the above result to each frequency component or to each sub-band of the signal. The result is that for those frequency bands, where the signal spectrum is below the noise spectrum no information needs to be transmitted to the receiver.

1.3.3.3 Predictive Coding with Specified Error Spectrum

In any speech coding system that adds noise to the speech signal, it is not sufficient to reduce the noise power; the goal is to minimize subjective loudness of the noise [14]. The subjective loudness of the noise is determined not just by its total power but also by the distribution of the noise and signal powers along the basilar membrane [15, 16].

The problem of encoding a source (original speech) with spectrum $S(f)$ and with error spectrum $E(f)$ is equivalent to the problem of coding a source "modified speech" with spectrum $E(f)/S(f)$ and a flat error spectrum [17]. The modified speech signal y_n is obtained by filtering the original speech signal through a filter with a magnitude-squared transfer function $1/E(f)$.

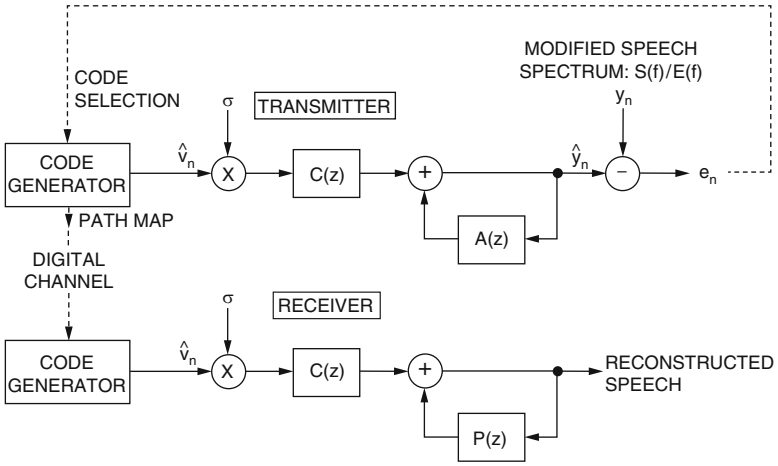

Fig. 1.2 Block diagram of a coder for a spectrally modified speech signal with white quantizing noise

The sequence of steps one might undertake to encode y_n is illustrated in Fig. 1.2. In the figure, $A(z)$ is the linear predictor for y_n and σ^2 is the mean-squared prediction error.

The code generator is assumed to be capable of generating all allowable codes from a zero-mean unit-variance white Gaussian process. A particular code is selected by a specified search procedure and the resulting sequence is scaled by a constant factor σ and filtered by two cascaded linear filters with transfer functions $C(z)$ and $[1-A(z)]^{-1}$, respectively. The filtered output is compared with the signal y_n to yield the error e_n. The optimal sequence is the one which minimizes the mean-squared error. Its "path map" is transmitted to the receiver, where the same sequence can be regenerated. The magnitude-squared transfer function of the filter $C(z)$ is given by [17, 18]

$$|C(z)|^2 = \max\left[0,\ 1 - \left(\theta/\sigma^2\right)|1 - A(z)|^2\right],$$

where θ is the minimum mean-squared error. Only in the limit of very high bit rates (i.e., $\theta \to 0$) is the filter not needed.

The minimization of subjective loudness of noise requires short-time spectral analysis of speech and noise; such computations can result in significant communication delays and therefore are not suitable for telephone applications. However, such computations can be carried out in audio coders that are used for efficient storage or for one-way transmission of signals. We still have to realize the desired noise spectrum $E(f)$ in speech coders. Communication delay is kept small by using recursive filters which can keep noise in frequency regions between the formants and harmonics at low levels [19].

1.3.3.4 Overcoming the Computational Complexity of Predictive Coders

For proper selection of codewords using a meaningful fidelity criterion in the coder shown in Fig. 1.2, the speech signal must be processed in blocks of approximately 5–10 ms in duration. For a sampling frequency of 8 kHz, a block of speech 10 ms in duration has 80 samples. To encode the prediction error at 1 bit/sample, the codebook has to be searched to find the best codeword out of 2^{80} codewords. This is impractical.

1.3.3.4.1 Multipulse Linear Predictive Coding

In our first attempt, we employed a sub-optimal procedure by building the codeword one "pulse" at a time, but still computing the mean-squared error over 10 ms. The location of each pulse was variable and was selected to minimize the total error over 10 ms interval. We called this coder a "multipulse linear predictive coder [20, 21] (multipulse LPC)". The locations and amplitudes of the pulses in the multipulse coder are obtained sequentially—one pulse at a time. After the first pulse has been determined, a new error is computed by subtracting out the contribution of this pulse to the error and the location of the next pulse is determined by finding the minimum of the new error. The process of locating new pulses is continued until the error is reduced to acceptable values or the number of pulses reaches the maximum value that can be encoded at the specified bit rate. We found that eight pulses in a 10 ms interval were sufficient for producing speech of high quality, regardless of whether speech was voiced or otherwise. We encoded the multipulse signal at a bit rate of 12 kb/s using run-length coding. Additional information around 4–6 kb/s was needed to code the predictor coefficients, resulting in a bit rate of about 16–18 kb/s for the speech signal. This was the first time we were able to produce high-quality natural-sounding speech (close in quality to 7-bit μ-law PCM) at these bit rates and the results were reported [20] at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in Paris in May 1982. The results of multipulse predictive coding convinced us that the techniques of predictive coding hold promise for providing superior performance at low bit rates.

1.3.3.4.2 Encoding with Binary Trees

We continued our work on speech coding to achieve high speech quality at even lower bit rates below 10 kb/s. We investigated binary tree encoding methods [22–24] with two branches from each node. A list of random Gaussian numbers were generated once and stored both at the transmitter and the receiver. The branches of the binary tree were "populated" with these numbers as needed in a sequential fashion. Thus, the first branch was populated with the first random number, the second branch with the second random number, and so on. This resulted in a 1 bit/sample coding of the prediction error. We will not discuss the details of the search strategy here, but the speech quality was high.

1.3.3.4.3 Encoding with Trees Populated with Block Codes

For bit rates lower than 1 bit/sample, it is necessary to populate each branch of the tree with a random sequence representing more than one sample of a codeword. The bit rate for a tree with B branches and N samples per branch is $(1/N)\ \log_2 B$ bit/sample. High quality speech was produced with a tree with ½ bit/sample (four branches and four samples per branch) at a bit rate of 4 kb/s for the prediction error [25].

1.3.3.4.4 Code-Excited Linear Predictive Coder (CELP)

In code-excited linear predictive coder, the set of possible codewords is stored in a code-book. For a given speech segment, the optimum codeword is selected to optimize a given fidelity criterion by exhaustive search of the codebook and an index specifying the optimum codeword is transmitted to the receiver. In general, such a search is impractical due to the large size of the codebooks. However, at very low bit rates, exhaustive search of the codebook becomes feasible [26].

Consider the coding of a short block of speech signal 5 ms in duration. Each such block consists of 40 speech samples at a sampling frequency of 8 kHz. A bit rate of ¼ bit per sample corresponds to 1024 possible sequences (10 bits) of length 40 for each block. The transmitter of a CELP coder is shown in Fig. 1.3. Each member of the codebook provides 40 samples of the prediction error and each sample is scaled by an amplitude factor (gain) that is constant for the 5 ms block and is reset to a new value once every 5 ms. The scaled samples are filtered sequentially through two recursive filters, long-delay and short-delay correlation filters. The regenerated speech samples at the output of the second filter are compared with

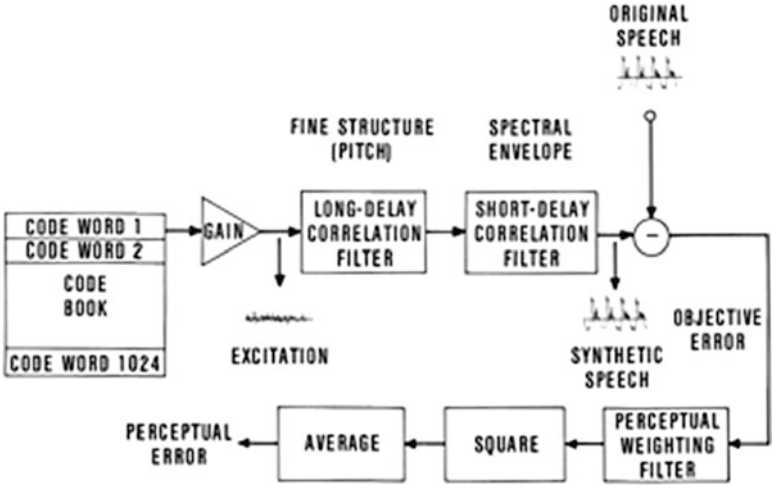

Fig. 1.3 Block diagram of the transmitter of a code-excited linear predictive coder for selecting the optimum codeword

the corresponding samples of the original speech signal to form a difference signal. The difference signal representing the objective error is further processed through a linear filter to attenuate those frequencies where the error is perceptually less important and to amplify those frequencies where the error is perceptually more important. Again, the codebook is populated with random sequences from a zero-mean unit-variance white Gaussian process.

Our research work in early 1980s with the code-excited linear predictive coder demonstrated that such coders offer considerable promise for producing high quality speech at bit rates as low as 4.8 kb/s. The search procedure was still computationally expensive; it took 125 s of Cray-1 CPU time to process 1 s of the speech signal. The Cray-1 time was charged by Bell Laboratories central computing center at $2,000 per hour; each second of processed speech on Cray-1 produced a charge of about $70 to our local budget and this was expensive research. The good news was that the first generation of DSP chips from Motorola, Texas Instruments, and Western Electric were already available and their prices were rapidly falling. Thanks to the Moore's law, performance of such chips was doubling every 18 months.

Looking back at 1985, 60 years had passed since Homer Dudley introduced vocoders for the first time and the dream of coding speech to reduce the bit rate by almost eight times over PCM at 64 kb/s had been realized. Low bit rate speech coders were ready for commercial application in the telephone system.

Several algorithms have been developed to provide reduction in complexity of the CELP coder. VSELP [27] is one such algorithm. The codebooks in the VSELP encoders are organized with a predefined structure, which significantly reduces the time required for search of the optimum codeword. VSELP coder with a bit rate of 8 kb/s was selected by TIA, the Telecommunications Industry Association, as the standard [27] for use in North American digital cellular telephone systems in 1989.

Algebraic Code Excited Linear Prediction [28] (ACELP) is another algorithm for reducing the search complexity of the CELP coders. This was the breakthrough that made CELP computationally tractable—the number of pulses needed was not great (4–8 per frame) and the pulse amplitudes could all be the same. This structure is fundamental to G.729 and all subsequent CELP codecs today. The name for the method comes from the fact that the pulse positions in the codewords are encoded to form an algebraic code. The benefit is that the codebook does not need to be stored, because it is algorithmically generated and even more importantly, it leads to an efficient search for the best codeword. Due to these benefits, ACELP has been used in several standards [29, 30].

1.4 Cellular Telephone Service

We are in the midst of a wireless revolution. The current revolution began with the birth of the "cellular concept" at Bell Laboratories in 1947. However, the technology to realize the concept did not exist. The cellular concept divided a service area into a number of smaller "cells". Radio frequencies could be reused in cells that were far enough apart.

In 1981, FCC released bandwidth in the 800–900 MHz range for commercial operation of cellular phone service in the United States. The first generation cellular systems were based on analog FM technology. AT&T started the first commercial cellular telephone system in the United States in Chicago in 1983.

To meet the rapid growth of demand for cellular service in North America, TIA, the Telecommunications Industry Association proposed in 1988 that the next generation of cellular systems provide a tenfold increase in capacity and use digital technology. Efforts to setup digital cellular telephony standards were started in United States, Europe and Japan. Standards were set for speech coders as well as for the multiple access technologies. Earlier analog FM systems used Frequency-Division Multiple Access (FDMA), but digital cellular systems use time-division multiple access (TDMA) or code-division multiple access (CDMA) technology.

1.4.1 Digital Cellular Standards

1.4.1.1 North American Digital Cellular Standards

In North America, the Telecommunication Industries Association (TIA) of the Electronic Industries Association (EIA) sets the standard for cellular communication [29]. TIA standardized IS-54 VSELP coder at a bit rate of 8 kb/s for use in North America in 1989. VSELP encodes speech at fixed bit rates and does not achieve more than threefold increase in the capacity over the analog FM cellular system. TIA proposed the use of CDMA in the United States.

TIA adopted QCELP developed by Qualcomm [31] for IS-96-A standard, operating at variable bit rates between 8 and 0.8 kb/s controlled by a rate determination algorithm. Subsequently, TIA standardized IS-127, the enhanced variable rate coder (EVRC), and IS-733 (QCELP) for personal communication systems, operating at variable bit rates between 14.4 and 1.8 kb/s. For North American TDMA standards, TIA standardized IS-641-A, based on ACELP, for enhanced full rate speech coding [30].

1.4.1.2 European Digital Cellular Standards

The European Telecommunications Standards Institute (ETSI) sets cellular standards in Europe. Within ETSI, Groupe Special Mobile (GSM) standardized RPE-LTP “Regular pulse Excitation with Long-Term Predictor” in 1987 [29]. This coder has a bit rate of 13 kb/s. ETSI has also standardized half-rate (5.6 kb/s) coder using VSELP, enhanced full rate (12.2 kb/s) coder using ACELP, and an adaptive multi-rate (AMR) coder operating at eight bit rates from 12.2 to 4.75 kb/s (using ACELP with four rates for the full-rate and four for the half-rate channels). The AMR coder provided enhanced speech quality under high radio interference and also increased battery life. The AMR codec is dominant today, although the wideband version is preferred.

1.5 The Future

Alexander Graham Bell was working on harmonic telegraph in 1875 to send eight telegraph messages on a single wire, because the demand was growing rapidly and it was expensive to install more telegraph lines. Over more than a century, we have progressed from wired telegraph and telephones to wireless cellular phones and from analog to digital networks, but the problem is the same. Growing demand to send more and more bits on present digital wireless networks continues. The mobile telephone service was introduced in 1983 so that people could use a single telephone number to talk, no matter where they were—the home, the car, or on the road. Soon wireless networks were carrying pictures and movies in addition to voice. Now, cellular phones or smart phones are always with us, keeping the wireless networks busy.

Technology for making computer chips run faster and faster involves shrinking the size of transistors. Over the years, the lateral dimensions in microelectronic circuits have been shrinking steadily, just as Moore's law predicted. It is now becoming harder and harder to achieve the doubling of chip performance every 18 months. Will Moore's law breakdown in the future? Will quantum considerations limit higher speeds? Will new quantum nanodevices take over?

A famous quote from Alexander Graham Bell, "When one door closes, another door opens. But we so often look so long and so regretfully upon the closed door, that we do not see the one which has been opened for us." Bell failed in his attempts to build a new telegraph but succeeded in opening the "telephone door"; Western Union continued to look for long at the closed "telegraph door". We do not know what the future will be but it will emerge out of a succession of closed and open doors. The future will belong to those who can navigate their way through the open doors.

References

1. C. Gray, *Reluctant Genius Alexander Graham Bell and the Passion for Invention* (Arcade Publishing, New York, 2006), pp. 42–43
2. T.A. Watson, The birth and babyhood of the telephone, in *Telephone Archive* (Information Department AT&T, 1937), pp. 14–17. www.telephonearchive.com
3. H. Dudley, The carrier nature of speech. Bell Syst. Tech. J. **19**, 495–513 (1940)
4. M.R. Schroeder, Vocoders: analysis and synthesis of speech. Proc. IEEE **54**, 720–734 (1966)
5. B. Gold, A history of vocoder research at Lincoln Laboratories. Lincoln Lab. J. **3**, 163–202 (1990)
6. M.R. Schroeder, Recent progress in speech coding at Bell Laboratories, in *Proc. III Intl. Cong. Acoust.,* Stuttgart, September 1959, pp. 202–210
7. P. Elias, Predictive coding – part I. IRE Trans. Inform. Theory **IT-1**(1), 16–23 (1955)
8. P. Elias, Predictive coding – part II. IRE Trans. Inform. Theory **IT-1**(1), 30–33 (1955)
9. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* (MIT Press, Cambridge, 1949)

10. C.E. Shannon, A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 379–423 and 623–656 (1948)
11. C.E. Shannon, Coding theorems for a discrete source with a fidelity criterion. IRE Conv. Rec. **7**, 142–163 (1959)
12. R.A. McDonald, P.M. Schultheiss, Information rates of Gaussian signals under criteria constraining the error spectrum. Proc. IEEE **52**, 415–416 (1964)
13. A.N. Kolmogorov, On the Shannon theory of information transmission in the case of continuous signals. Proc. IEEE **52**, 415–416 (1964)
14. B.S. Atal, M.R. Schroeder, Predictive coding of speech signals and subjective error criteria. IEEE Trans. Acoust. Speech Signal Proc. **ASSP-27**, 247–254 (1979)
15. M.R. Schroder, B.S. Atal, J.L. Hall, Optimizing digital speech coders by exploiting masking properties of the human ear. J. Acoust. Soc. Am. **66**, 1647–1652 (1979)
16. M.R. Schroder, B.S. Atal, J.L. Hall, Objective measures of certain speech signal degradations based on properties of human auditory perception, in *Frontiers of Speech Communication Research,* ed. by B. Lindblom, S. Ohman (Academic Press, London, 1979), pp. 217–229
17. M.R. Schroeder, B.S. Atal, Rate distortion theory and predictive coding, in *Proceedings of International Conference on Acoustics Speech, and Signal Processing,* 201–204 (1981)
18. T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression* (Prentice-Hall, Englewood Cliffs, 1971), pp. 235–239
19. B.S. Atal, Predictive coding of speech at low bit rates. IEEE Trans. Commun. **COM-30**, 600–614 (1982)
20. M.R. Schroeder, B.S. Atal, J.R. Remde, A new model of LPC excitation for producing natural-sounding speech, in *Proceedings of International Conference on Acoustics, Speech and Signal Processing,* 614–617 (1982)
21. B.S. Atal, High-quality speech at low bit rates: multi-pulse and stochastically-excited linear predictive coders, in *Proceedings of International Conference on Acoustics Speech, and Signal Processing,* 1681–1684 (1986)
22. J.B. Anderson, J.B. Bodie, Tree encoding of speech. IEEE Trans. Inform. Theory **IT-21**, 379–387 (1975)
23. F. Jelinek, Tree encoding of memoryless time-discrete sources with a fidelity criterion. Trans. IEEE **IT-15**, 584–590 (1969)
24. D.W. Becker, A.J. Viterbi, Speech digitization and compression by adaptive predictive coding with delayed decision, Conference Record of National Telecommunications Conference, vol. 2 (A77-15115 04-32) (Institute of Electrical and Electronics Engineers, Inc., New York, 1975), pp. 46-18–46-23
25. M.R. Schroeder, B.S. Atal, Speech coding using efficient block codes, in *Proceedings of International Conference* on *Acoustics, Speech and Signal Processing,* 1668–1671 (1982)
26. M.R. Schroeder, B.S. Atal, Code-excited linear prediction (CELP): high quality speech at very low bit rates, in *Proceedings of International Conference* on *Acoustics, Speech and Signal Processing,* 937–940 (1985)
27. I.A. Gerson, M.A. Jesuik, Vector sum excited linear prediction analysis (VSELP), in *Advances in Speech Coding*, ed. by B.S. Atal, V. Cuperman, A. Gersho (Kluwer Academic Publishers, Boston, 1991), pp. 69–79
28. J.-P. Adoul, P. Mabilleau, M. Delprat, S. Morissette, Fast CELP coding based on algebraic codes, in *Proceedings of International Conference on Acoustics, Speech and Signal Processing,* 1957–1960 (1987)
29. R.V. Cox, Speech coding standards, in *Speech Coding and Synthesis*, ed. by W.B. Kleijn, K.K. Paliwal (Elsevier, Amsterdam, 1995), pp. 49–78
30. A.M. Kondoz, *Digital Speech, Coding for Low Bit Rate Communication Systems* (Wiley, Hoboken, 2004), pp. 12–18
31. W. Gardner, P. Jacobs, C. Lee, QCELP: variable rate speech coder for CDMA digital cellular, in *Speech and Audio Coding for Wireless and Network Applications*, ed. by B.S. Atal, V. Cuperman, A. Gersho (Kluwer Academic Publishers, Boston, 1993), pp. 85–92

Chapter 2
Challenges in Speech Coding Research

Jerry D. Gibson

Abstract Speech and audio coding underlie many of the products and services that we have come to rely on and enjoy today. In this chapter, we discuss speech and audio coding, including a concise background summary, key coding methods, and the latest standards, with an eye toward current limitations and possible future research directions.

2.1 Introduction

We distinguish between speech and audio coding according to the bandwidth occupied by the input source. *Narrowband* or telephone bandwidth speech occupies the band from 200 to 3,400 Hz, and is the band classically associated with telephone quality speech. The category of *wideband* speech covers the band 50 Hz–7 kHz. Audio is generally taken to cover the range of 20 Hz–20 kHz, and this bandwidth is sometimes referred to today as *fullband* audio [1, 2]. In recent years, quite a few other bandwidths have attracted attention, primarily for audio over the Internet applications, and the bandwidth of 50 Hz–14 kHz, designated as *superwideband*, has gotten considerable recent attention [3]. As the frequency bands being considered move upward from narrowband speech through wideband speech and superwideband audio, on up to fullband audio, the basic structures for digital processing and the quality expectations change substantially. In the following, we elaborate on these differences and highlight the challenges in combining the processing of this full range of bandwidths in single devices.

J.D. Gibson (✉)
Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA 93106-6065, USA
e-mail: gibson@ece.ucsb.edu

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_2

2.2 Speech Coding

The goal of speech coding is to represent speech in digital form with as few bits as possible while maintaining the intelligibility and quality required for the particular application [2]. Speech coding is a critical technology for videoconferencing systems, digital cellular communications, and voice over Internet protocol (VoIP) applications, while audio coding is essential for portable audio players, audio streaming, video streaming, and the storage and playback of movies.

The basic approaches for coding narrowband speech evolved over the years from waveform following codecs to code excited linear prediction (CELP) based codecs [2]. The process of this evolution was driven by applications that required lower bandwidth utilization and by advances in digital signal processing, which were facilitated by improvements in processor speeds that allowed more sophisticated processing to be incorporated. The reduction in bit rates was obtained by relaxing constraints on encoding delay and on complexity. This later relaxation of constraints, particularly on complexity, should be a lesson learned for future research; namely, complexity should not be a dominating concern at the beginning of a basic research effort.

Note that the basic speech coding problem for narrowband speech, in particular, follows the *distortion rate* paradigm; that is, given a rate constraint set by the application, the codec is designed to minimize distortion. The resulting distortion is not necessarily small or inaudible—just acceptable for the given constraints. The distortion rate structure should be contrasted with the *rate distortion* problem wherein the constraint is on allowable distortion and the rate required to achieve that distortion is minimized. Notice that for the rate distortion approach, a specified distortion is the goal and the rate is adjusted to obtain this level of distortion. Voice coding for digital cellular communications is an example of the distortion rate approach, since it has a rate constraint, while coding of fullband audio typically has the goal of transparent quality, and hence is an example of the rate distortion paradigm. We elaborate more on these ideas in the following.

We use the terms speech coding and voice coding interchangeably in this paper. Generally, it is desired to reproduce the voice signal, since we are interested in not only knowing what was said, but also in being able to identify the speaker.

Given a particular source such as voice, audio, or video, the classic tradeoff in lossy source compression is rate versus distortion—the higher the rate, the smaller the average distortion in the reproduced signal. Of course, since a higher bit rate implies a greater channel or network bandwidth requirement, the goal is always either to minimize the rate required to satisfy the distortion constraint or minimize the distortion for a given rate constraint. For speech coding, we are interested in achieving a quality as close to the original speech as possible within the rate, complexity, latency, and any other constraints that might be imposed by the application of interest. Encompassed in the term quality are intelligibility, speaker identification, and naturalness. Absolute category rating (ACR) tests are subjective tests of speech quality and involve listeners assigning a category and rating for each

speech utterance according to the classifications, such as, Excellent (5), Good (4), Fair (3), Poor (2), and Bad (1). The average for each utterance over all listeners is the Mean Opinion Score (MOS) [1].

Of course, listening tests involving human subjects are difficult to organize and perform, so the development of objective measures of speech quality is highly desirable. The perceptual evaluation of speech quality (PESQ) method, standardized by the ITU-T as P.862, was developed to provide an assessment of speech codec performance in conversational voice communications. The PESQ has been and can be used to generate MOS values for both narrowband and wideband speech [4, 5]. While no substitute for actual listening tests, the PESQ and its wideband version are widely used for initial codec evaluations and are highly useful. A newer objective measure, designated as P.863 POLQA (Perceptual Objective Listening Quality Assessment) has been developed but it has yet to receive widespread acceptance [6]. For a tutorial development of perceptual evaluation of speech quality, see [7].

More details on MOS and perceptual performance evaluation for voice codecs are provided in the references [1, 2, 7]. Later in the chapter, we discuss the relatively new nine point ACR ratings that are becoming popular as superwideband speech and audio become more prevalent in codec designs.

2.2.1 Speech Coding Methods

The most common approaches to narrowband speech coding today center around two paradigms, namely, waveform-following coders and analysis-by-synthesis methods. Waveform-following coders attempt to reproduce the time domain speech waveform as accurately as possible, while analysis-by-synthesis methods utilize the linear prediction model and a perceptual distortion measure to reproduce only those characteristics of the input speech determined to be most important perceptually. Another approach to speech coding breaks the speech into separate frequency bands, called subbands, and then codes these subbands separately, perhaps using a waveform coder or analysis-by-synthesis coding for each subband, for reconstruction and recombination at the receiver. Extending the resolution of the frequency domain decomposition leads to transform coding and coding using filter banks, wherein a transform is performed on a frame of input speech/audio and the resulting transform coefficients are quantized and transmitted to reconstruct the speech/audio from the inverse transform. Subband decompositions and transform based decompositions are very closely related and combinations of the two are common in codecs that code bandwidths beyond narrowband speech.

2.2.1.1 Waveform Coding [2]

Familiar waveform-following methods are logarithmic pulse code modulation (log-PCM) and adaptive differential pulse code modulation (ADPCM), and both have

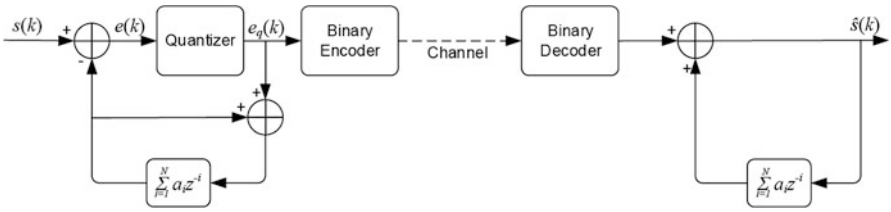

Fig. 2.1 An ADPCM encoder and decoder

found widespread applications. Log PCM at 64 kilobits/s (kbps) is the speech codec that was the work horse for decades in the long distance public switched telephone network at a rate of 64 kbps, and it is the most widely employed codec for VoIP applications. It is a simple coder and it achieves what is called toll quality, which is the standard level of performance against which all other narrowband speech coders are judged.

Log PCM uses a nonlinear quantizer to reproduce low amplitude signals, which are important to speech perception, well. There are two closely related types of log-PCM quantizer used in the World—μ-law, which is used in North America, South Korea and Japan, and A-law, which is used in the rest of the world. Both achieve toll quality speech, and which, in terms of the MOS value is usually between 4.0 and 4.5 for log-PCM. These quality levels are considered very good but not transparent.

ADPCM operates at 40 kbps or lower, and it achieves performance comparable to log-PCM by using an adaptive linear predictor to remove short-term redundancy in the speech signal before adaptive quantization of the prediction error. See Fig. 2.1. The reasoning behind differential coding like ADPCM is that by subtracting a predicted value from each input sample, the dynamic range of the signal to be quantized is reduced, and hence, good reproduction of the signal is possible with fewer bits. The most common form of ADPCM uses what is called backward adaptation of the predictors and quantizers to follow the waveform closely. Backward adaptation means that the predictor and quantizer are adapted based upon past reproduced values of the signal that are available at the encoder and decoder [2]. No predictor or quantizer parameters are sent along as side information with the quantized waveform values.

2.2.1.2 Subband and Transform Methods [2]

The process of breaking the input speech into subbands via bandpass filters and coding each band separately is called subband coding. To keep the number of samples to be coded at a minimum, the sampling rate for the signals in each band is reduced by decimation. Of course, since the bandpass filters are not ideal, there is some overlap between adjacent bands and aliasing occurs during decimation. Ignoring the distortion or noise due to compression, Quadrature mirror filter (QMF) banks allow the aliasing that occurs during filtering and subsampling at the encoder

to be cancelled at the decoder. The codecs used in each band can be PCM, ADPCM, or even an analysis-by-synthesis method. The advantage of subband coding is that each band can be coded to a different accuracy and that the coding error in each band can be controlled in relation to human perceptual characteristics.

Transform coding methods were first applied to still images but later investigated for speech. The basic principle is that a block of speech samples is operated on by a discrete unitary transform and the resulting transform coefficients are quantized and coded for transmission to the receiver. Low bit rates and good performance can be obtained because more bits can be allocated to the perceptually important coefficients, and for well-designed transforms, many coefficients need not be coded at all, but are simply discarded, and acceptable performance is still achieved.

Although classical transform coding has not had a major impact on narrowband speech coding and subband coding has fallen out of favor in recent years (with a slight recent resurgence such as the adoption of a subband codec optional for Bluetooth [8]), filter bank and transform methods play a critical role in high quality audio coding, and several important standards for wideband, superwideband, and fullband speech/audio coding are based upon filter bank and transform methods. Although it is intuitive that subband filtering and discrete transforms are closely related, by the early 1990s, the relationships between filter bank methods and transforms were well-understood [9]. Today, the distinction between transforms and filter bank methods is somewhat blurred, and the choice between a filter bank implementation and a transform method may simply be a design choice. Often a combination of the two is the most efficient.

2.2.1.3 Analysis-by-Synthesis Methods [2, 10]

Analysis-by-synthesis (AbS) methods are a considerable departure from waveform-following techniques and from frequency domain methods as well, although they do build on linear prediction as used in ADPCM. The most common and most successful analysis-by-synthesis method is code-excited linear prediction (CELP). In CELP speech coders, a segment of speech (say, 5–10 ms) is synthesized using the linear prediction model along with a long-term redundancy predictor for all possible excitations in what is called a codebook. For each excitation, an error signal is calculated and passed through a perceptual weighting filter.

This operation is represented in Fig. 2.2a. The excitation that produces the minimum perceptually weighted coding error is selected for use at the decoder as shown in Fig. 2.2b. Therefore, the best excitation out of all possible excitations for a given segment of speech is selected by synthesizing all possible representations at the encoder, hence, the name analysis-by-synthesis (AbS). The predictor parameters and the excitation codeword are sent to the receiver to decode the speech. It is instructive to contrast the AbS method with waveform coders such as ADPCM where each sample is coded as it arrives at the coder input.

The perceptual weighting is key to obtaining good speech coding performance in CELP, and the basic idea is that the coding error is spectrally shaped to fall below

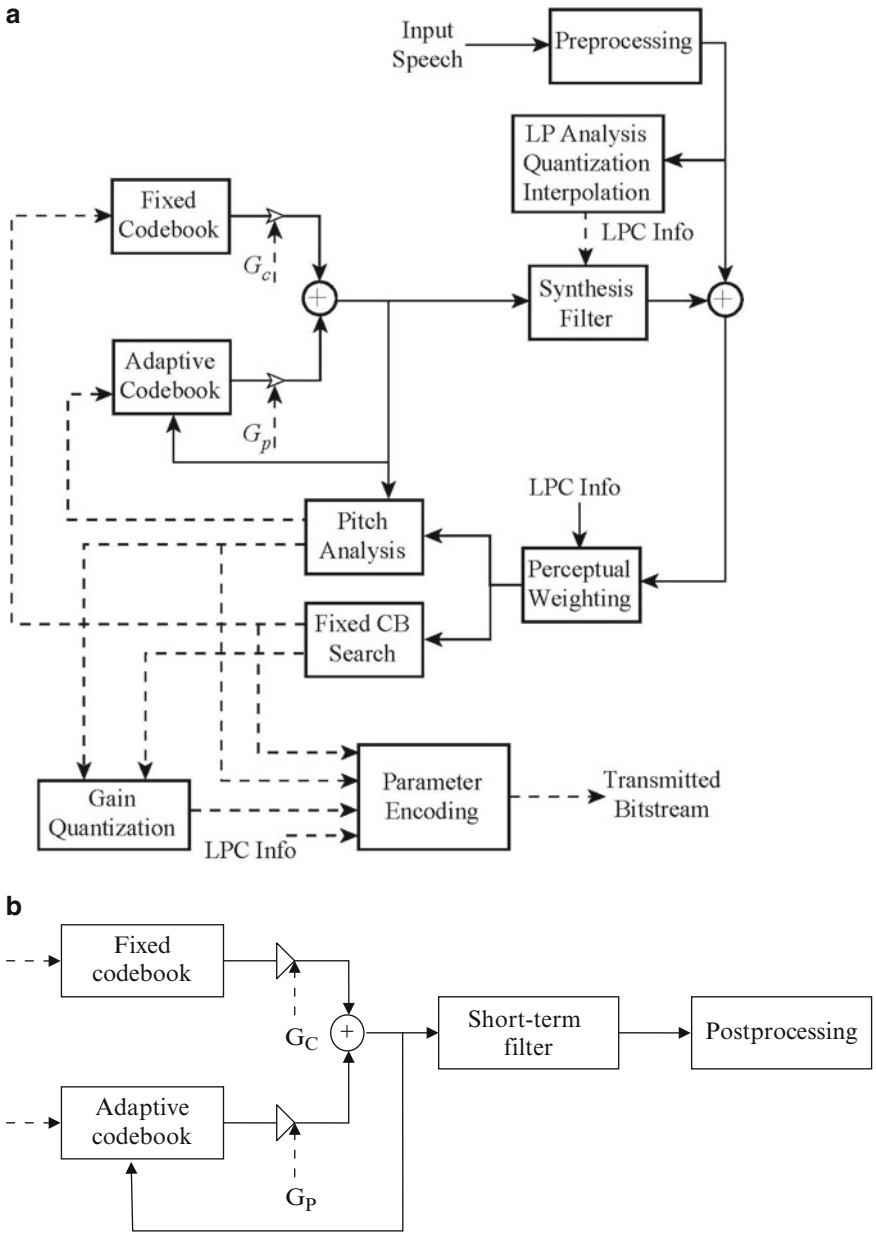

Fig. 2.2 (**a**) Encoder for code-excited linear predictive (CELP) coding with an adaptive codebook. (**b**) CELP decoder with an adaptive codebook and postfiltering

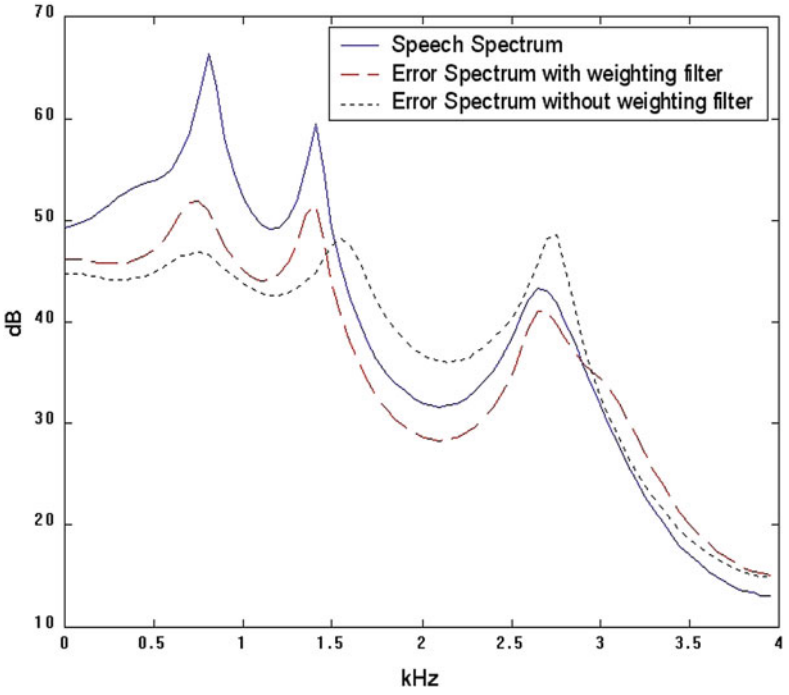

Fig. 2.3 Perceptual weighting of the coding error as a function of frequency

the envelope of the input speech across the frequency band of interest. Figure 2.3 illustrates the concept wherein the spectral envelope of a speech segment is shown, along with the coding error spectrum without perceptual weighting (unweighted denoted by short dashes) and the coding error spectrum with perceptual weighting (denoted by long dashes). The perceptually weighted coding error falls below the spectral envelope of the speech across most of the frequency band of interest, just crossing over around 3,100 Hz. The coding error is thus masked by the speech signal itself. In contrast, the unweighted error spectrum is above the speech spectral envelope starting at around 1.6 kHz, which produces audible coding distortion for the same bit rate. The reader should note that if one analyzes each frame of a CELP-coded speech segment, the goal of pushing the error spectrum below that of the input speech is often not obtained across the entire band. This is because the perceptually shaping methods used are approximate and have not yet been refined to guarantee the desired result [2].

In recent years, it has become common to use an adaptive codebook structure to model the long term memory rather than a cascaded long term predictor. A decoder using the adaptive codebook approach is shown in Fig. 2.2b. The analysis-by-synthesis procedure is computationally intensive, and it is fortunate that algebraic codebooks, which have mostly zero values and only a few nonzero pulses, have been discovered and work well for the fixed codebook [10].

2.2.1.4 Postfiltering [11]

Although a perceptual weighting filter is used inside the search loop for the best excitation in the codebook for analysis-by-synthesis methods, there is often some distortion remaining in the reconstructed speech that is sometimes characterized as "roughness." This distortion is attributed to reconstruction or coding error as a function of frequency that is too high at regions between formants and between pitch harmonics. Several codecs thus employ a postfilter that operates on the reconstructed speech to de-emphasize the coding error between formants and between pitch harmonics. This is shown as "Post-processing" in Fig. 2.2b.

The general frequency response of the postfilter has the form similar to the perceptual weighting filter with a pitch or long term postfilter added. There is also a spectral tilt correction since the formant-based postfilter results in an increased low pass filter effect, and a gain correction term [2, 10, 11]. The postfilter is usually optimized for a single stage encoding (however, not always), so if multiple tandem connections of speech codecs occur, the postfilter can cause a degradation in speech quality.

2.2.1.5 Voice Activity Detection and Silence Coding

For many decades, researchers have been interested in assigning network capacity only when a speaker is "active," by removing silent periods in speech to reduce the average bit rate. This was successfully accomplished for some digital cellular coders where silence is removed and coded with a short length code and then replaced at the decoder with "comfort noise." Comfort noise is needed because the background sounds for speech coders are seldom pure silence and inserting pure silence generates unwelcome artifacts at the decoder and can cause the impression that the call is lost [10].

Today, many codecs use voice activity detection to excise non-speech signals so that non-speech regions do not need to be coded explicitly. More sophisticated segmentation can also be performed so that different regions can be coded differently. For example, more bits may be allocated to coding strongly voiced segments and fewer allocated to unvoiced speech. Also, speech onset might be coded differently as well.

2.2.2 Speech Coding Standards

Different standardization bodies have adopted a host of codecs for rapidly evolving applications. For narrowband speech coding, the ITU-T and the several digital cellular standardization efforts are the dominant activities. There is a vast number of standards that have been set. We begin the discussion with ITU-T standardized codecs since some of those codecs have served as the basis for cellular codecs, and since some of these codecs have been adopted for VoIP applications.

Table 2.1 Comparison of ITU-T narrowband speech codecs

Standards body	ITU			
Recommendation	G.711	G.726	G.728	G.729
Coder type	Companded PCM	ADPCM	LD-CELP	CS-ACELP
Bite rate (kbps)	64	16–40	16	8
Complexity (MIPS)	$<<1$	~ 1	~ 30	≤ 20
Frame size (ms)	0.125	0.125	0.625	10
Lookahead (ms)	0	0	0	5
Codec delay (ms)	0.25	0.25	1.25	25

2.2.2.1 ITU-T Standards

Table 2.1 lists some of the narrowband voice codecs that have been standardized by the ITU-T over the years, including details concerning the codec technology, transmitted bit rate, performance, complexity, and algorithmic delay. Those shown include G.711, G.726, G.728, and G.729 for narrowband (telephone bandwidth) speech (200–3,400 Hz), where the first two codecs are waveform-following codecs, and the latter three are variations on code excited linear prediction.

G.711 at 64 kilobits/s (kbps) is the voice codec most often used in VoIP and many applications wherein somewhat higher bit rates are workable and very low complexity is desirable. This codec is based on a nonlinear scalar quantization method called logarithmic pulse code modulation (log-PCM), as discussed earlier. The G.726 waveform-following codec is based on ADPCM and operates at bit rates of 40, 32, 24, and 16 kbps. This codec achieves low delay and is still considered a low complexity codec. G.728 is a code-excited technique but when it was standardized, it was still desired to have a low encoding delay of 5 ms or less. G.728 is much more complex than either G.726 or G.711.

As the desired bit rate moved toward 8 kbps, the low delay requirement was relaxed. This allowed code-excited linear prediction methods to move to the forefront. The G.729 codec is an analysis-by-synthesis codec based on algebraic code excited linear prediction (ACELP), and it uses an adaptive codebook to incorporate the long term pitch periodicity [2, 10]. In addition to a lower complexity version of G.729, called G.729A, there is a higher rate codec based on G.729, designated G.729E, and a wideband version designated G.729.1 [12]. The G.729 codec structure has been very influential on subsequent voice coding standards for VoIP and digital cellular networks and this structure can be seen in most standardized voice codecs today.

Even though we are quite comfortable communicating using telephone bandwidth speech (200–3,400 Hz), there is considerable interest in compression methods for wideband speech covering the range of 50 Hz–7 kHz. The primary reasons for the interest in this band are that wideband speech (and wider bands) improves intelligibility, naturalness, and speaker identifiability. Table 2.2 lists codecs for

Table 2.2 ITU-T wideband and fullband speech coding standards

Recommendation	ITU-T G.722	ITU-T G.722.1	ITU-T G. 722.2 3GPP AMR-WB	ITU-T G.718	ITU-T G.719
Coder type	Sub-band ADPCM	MLT	ACELP	ACELP, MDCT	Adaptive resolution MDCT, FLVQ
Audio bandwidth (Hz)	50–7,000	50–7,000	50–7,000	50–7,000	20–20,000
Bitrate(s) (kbits/s)	48, 56, 64	24, 32	6.6, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85, 23.05, 23.85	8, 12, 16, 24, 32 & 12.65 (G.722.2, AMR-WB, VMR-WB Interop Mode)	32 . . . 128 steps of 4 kbps up to 96 kbps, steps of 8 kbps up to 128 kbps
Frame length (ms)	0.125	20	20	20	20
Algorithmic delay (ms)	1.625	40	25	32.875–43.875	40
Comp. complexity	10 MIPS	<5.5 WMOPS	27.2–39.0 WMOPS	57 WMOPS	15.39–21 WMOPS

wideband speech, including G.722, G.722.1 [13], and G.722.2 [14]. Also, shown in the table are ITU-T codecs G.718 for wideband speech (50 Hz–7 kHz) [15], and G.719 for fullband audio [16, 17].

The first application of wideband speech coding was to videoconferencing, and the first standard, G.722, separated the speech into two subbands and used ADPCM to code each band. The G.722 codec is relatively simple and produces good quality speech at 64 kbps, and lower quality speech at the two other possible codec rates of 56 and 48 kbps [2]. G.722 at 64 kbps is often employed as a benchmark for the performance of other wideband codecs.

Two additional wideband speech coding standards, designated as G.722.1 and G.722.2, utilize coding methods that are quite different from G.722, as well as completely different from each other. The G.722.1 standard employs a filter bank/transform decomposition called the modulated lapped transform (MLT) and operates at the rates of 24 and 32 kbps. The coder has an algorithmic delay of 40 ms, which does not include any computational delay. Since G.722.1 employs filter bank methods, it performs well for music and less well for speech. This codec structure for G.722.1 has much in common with the fullband audio codecs used for many music player products such as MP3.

G.722.2 is an ITU-T designation for the adaptive multirate wideband (AMR-WB) speech coder standardized by the cellular body 3GPP [14]. This coder operates at rates of 6.6, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85, 23.05, and 23.85 kbps and is based upon an algebraic CELP (ACELP) analysis-by-synthesis codec. Since ACELP utilizes the linear prediction model, the coder works well for speech but less well for music, which does not fit the linear prediction model. G.722.2 achieves good speech quality at rates greater than 12.65 kbps and performance equivalent to G.722 at 64 kbps with a rate of 23.05 kbps and higher.

G.718 is a wideband speech codec that has an embedded codec structure and that operates at 8, 12, 16, 24, and 32 kbps, plus a special alternate coding mode that is bit stream compatible with AMR-WB. G.719 is a fullband audio codec that has relatively low complexity and low delay for a fullband audio codec, and the complexity is approximately evenly split between the encoder and decoder. This codec is targeted toward real-time communications such as in videoconferencing systems and the high definition telepresence applications.

2.2.2.2 Digital Cellular Standards

Digital cellular applications impose a stringent set of requirements on voice codecs in addition to rate and quality, such as complexity, robustness to background impairments, and the ability to perform well over wireless channels. Over the years, standards have been set by different bodies for different segmentations of the market, particularly according to geographic regions and wireless access technologies. More specifically, digital cellular standards were produced in the late

Table 2.3 Selected GSM voice codecs

Codec	Speech coding bit-rate (in kbit/s)	System/traffic channel	Speech coding algorithm	Complexity WMOPS
FR codec	13.0	GSM FR	Regular Pulse Excitation-Long Term Prediction (RPE-LTP)	3.0
HR codec	5.6	GSM HR	Vector-Sum Excited Linear Prediction (VSELP)	18.5
EFR codec	12.2	GSM FR	Algebraic Code Excited Linear Prediction (ACELP)	15.2
AMR codec	12.2, 10.2, 7.95, 7.4, 6.7, 5.9, 5.15, 4.75	GSM FR (all eight modes), GSM HR (six lowest modes), 3G WCDMA (all modes)	Algebraic Code Excited Linear Prediction (ACELP)	16.8
AMR-WB codec	23.85, 23.05, 19.85, 18.25, 15.85, 14.25, 12.65, 8.85, 6.60	GSM FR (seven lowest modes), EDGE (all modes), 3G WCDMA (all modes)	Algebraic Code Excited Linear Prediction (ACELP)	35.4

1980s and early 1990s in Europe, Japan, and North America. The competing North American standards then led to standards efforts more pointed toward each of the competing technologies.

The GSM standards developed in Europe were the basis of perhaps the first widely implemented digital cellular systems. Table 2.3 lists voice codecs standardized for GSM systems, wherein FR stands for "Full rate" and HR stands for "Half rate." The terms FR and HR refer to the total transmitted bit rate for combined voice coding and error correction (or channel) coding, and FR is always 22.8 kbps and HR is always 11.4 kbps. By subtracting the rate of the voice codec from either 22.8 or 11.4, one obtains the bit rate allocated to error control coding.

The first GSM FR voice codec standardized in 1989 was not an analysis-by-synthesis codec but used a simpler regular pulse excited linear predictive structure with a long term predictor. As a result, the codec had to be operated at 13 kbps to achieve the needed voice quality, but it had very low complexity. An important and somewhat dominant voice codec in recent years is the Adaptive Multirate Codec, both narrowband and wideband versions. Note that AMR-NB has multiple rates and can be operated as a FR or HR codec, depending upon the rates. For GSM, the AMR-NB codec rates are not source-controlled as some prior codecs were, but the rates are switchable and usually adjusted by the network. The AMR codec maintains compatibility with other systems by incorporating the GSM EFR (Enhanced Full Rate) codec at 12.2 kbps and IS-641 at 7.4 kbps as two of its selectable rates. The AMR wideband codec, AMR-WB, also based upon ACELP is also a very important codec today; however, note how the complexity has grown.

2.2.2.3 VoIP Standards

VoIP for wireless access points involves many of the same issues as for wireline VoIP, such as voice quality, latency, jitter, packet loss performance, and packetization. One new challenge that arises is that since the physical link in Wi-Fi is wireless, bit errors commonly occur and this, in turn, affects link protocol design and packet loss concealment. A second challenge is that congestion can play a role, thus impacting real time voice communications. The way these two issues relate to voice codecs are that packet loss concealment methods are more critical and that codec delay should be more carefully managed for such wireless access points.

Turning our attention to the voice codecs normally implemented in VoIP solutions, we find that at this point in time, many VoIP codecs are borrowed from other standards bodies. Specifically, G.711, G.729, and G.722 are commonly offered in VoIP products. Additionally, AMR-NB and perhaps AMR-WB are optional voice codecs. All of these codecs have well-developed packet loss concealment methods, which makes them quite compatible with wireless applications. One thing to notice is that the AMR codecs are the only ones that are common with any digital cellular standards, and this can lead to tandem coding penalties when digital cellular and wireless VoIP are used for portions of the same connection for a voice call. The need to support multiple codecs can also be an issue as cell phones morph into smartphones that support both digital cellular and wireless access point connectivity.

There have also been voice codecs developed outside of standards bodies and offered as open source. Two such codecs are Speex [18] and iLBC (internet Low Bitrate Codec) [19]. Speex has become obsolete with the introduction of the Opus codec [20, 21], described in a later section. These codecs have been compared to other standardized codecs in several studies [22–24].

2.3 Audio Coding [25, 26]

The basic very successful paradigm for audio coding, meaning coding full band audio, in the past two decades has been the filter bank/transform based approach with noise masking using an iterative bit allocation. This technique does not lend itself to real time communications directly because of the iterative bit allocation method and because of complexity, and to a lesser degree, delay in the filter bank/transform/noise masking computations. As a result, the primary impact of high quality audio coding has been to audio players (decoders) such as MP3 and audio streaming applications.

A high level block diagram of an audio codec is shown in Fig. 2.4. In this diagram, two paths are shown for the sampled input audio signal, one path is through the filter bank/transform that performs the analysis/decomposition into spectral components to be coded, and the other path into the psychoacoustic analysis that computes the noise masking thresholds. The noise masking thresholds are then

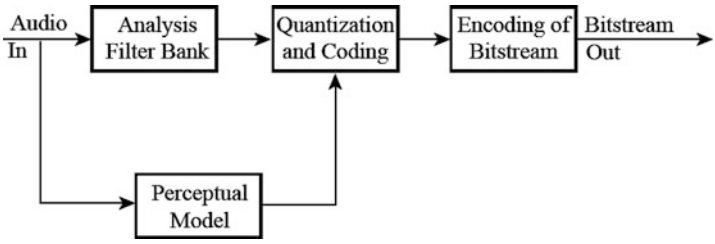

Fig. 2.4 Generic audio coding approach

used in the bit allocation that forms the basis for the quantization and coding in the analysis/decomposition path. All side information and parameters required for decoding are finally losslessly coded for storage or transmission.

The primary differences among the different audio coding schemes that have been standardized and/or found wide application are in the implementations of the time/frequency analysis/decomposition in terms of the types of filter banks/transforms used and their resolution in the frequency domain. Note that the frequency resolution of the psychoacoustic analysis is typically finer than the analysis/decomposition path since the perceptual noise masking is so critical for good quality. There are substantive differences in the other blocks as well, with many refinements over the years [2, 25, 26].

The strengths of the basic audio coding approach are that it is not model based, as in speech coding using linear prediction, and that the perceptual weighting is applied on a per-component basis, whereas in speech coding, the perceptual weighting relies on a spectral envelope shaping. A weakness in the current approaches to audio coding is that the noise masking theory that is the foundation of the many techniques is three decades old; further, the masking threshold for the entire frame is computed by adding the masking thresholds for each component. The psychoacoustic/audio theory behind this technique of adding masking thresholds has not been firmly established [25].

Other key ideas in the evolution of the full band audio coding methods have been pre- and post-masking and window switching to capture transients and steady state sounds. Details of the audio coding methods are left to the very comprehensive references cited [25, 26]. However, we will revisit full band audio coding in discussing the newer standards and when considering new research directions.

2.4 Newer Standards

The standardization processes continue to be vigorous in the classically active standards bodies such as ITU-T, the ISO, and the digital cellular community. Furthermore, there is considerable activity in developing alternative coding methods outside of the standards bodies that may be free of intellectual property claims.

A unifying thread in all of these efforts to develop new codecs is to have codecs that cover narrowband, wideband, superwideband, and full band in one specification. Differences in the codec development efforts revolve around whether the delay needed in coding is low enough to allow real time communications or whether the latency precludes most such real time applications. See [1, 27] for a more complete discussion of delay in voice codec design and its impact on voice coding applications.

Efforts in the last decade to design one codec that would cover all bands from narrowband speech to fullband audio have led to the approach of combining code excited linear prediction methods to cover narrowband and wideband speech with the filter bank methods to cover full band audio, using switching or mixing in between. Such a codec structure may be called an integrated codec, and examples of such integrated codecs include G.718 which combines ACELP and MDCT technologies, originally only covering 50 Hz–7 kHz but since extended to superwideband [28], and the recently standardized MPEG USAC (Unified Speech and Audio Coding) architecture shown in Fig. 2.5, which covers the entire range from 20 Hz to 20 kHz, with the goal of coding voice and fullband audio well [29]. The USAC codec utilizes signal classification and down mixes the stereo to mono for coding in the low band. There is a low pass/high pass decomposition, and enhanced spectral band replication (eSBR) is used to code the high band. There is both a baseline mode and an extension mode.

The applications targeted for the MPEG USAC codec are multimedia download to mobile devices, audio books, mobile TV, and digital radio. While there are strong targets for improving audio performance and it is designed to code speech, audio, and mixed content, there are no specifications on complexity or delay.

At first inspection, these integrated structures can be viewed as merely bolting together successful codecs for different bands, but the USAC effort notes that it is the handling of the transitions between different coding paradigms that requires innovation beyond a simple combination of known schemes. It is not difficult to understand how challenging it is to combine such different codec designs, and so the USAC and related codecs must be considered a substantial advance in the state-of-the-art.

A key limitation of the USAC effort is the lack of support for conversational services, which require low encoding delay and limitations on complexity. Several new and developing standards try to encompass these very important conversational applications.

For conversational speech, the ITU-T standardization efforts have already resulted in G.711.1 [30] and G.729.1 [12], both of which are extensions of existing standards to wider bands and different, higher rates. A superwideband version of G.722.1, designated G.722.1 Annex C [31], has also been standardized. The G.722.1C codec has a coding delay of 40 ms and a relatively modest complexity with transmitted bit rates of 24, 32 and 48 kbps. It codes voice, audio, and natural sounds well, and it is targeted for applications to videoconferencing, VoIP, and battery powered devices.

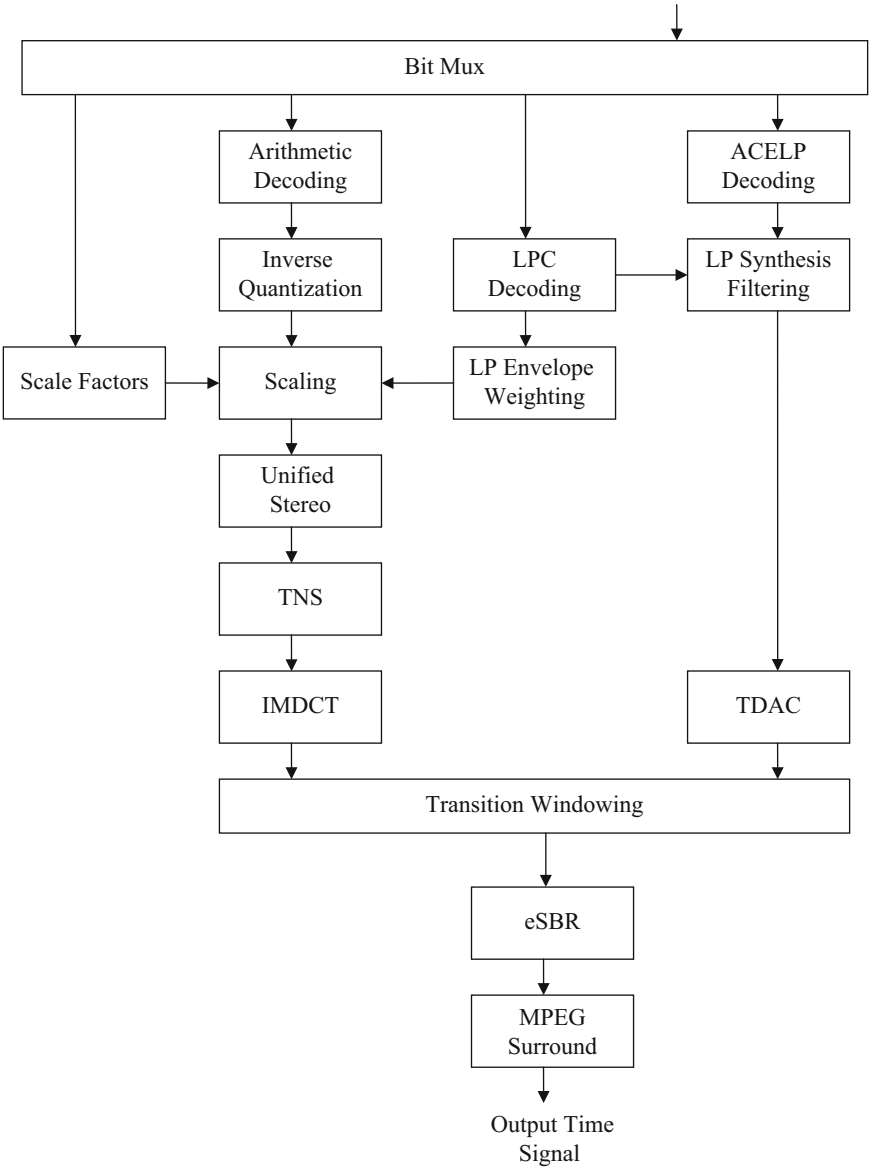

Fig. 2.5 USAC decoder structure

It is evident that digital cellular systems Worldwide will be based on 3GPP Long Term Evolution (LTE), which utilizes Orthogonal Frequency Division Multiple Access (OFDMA) in the downlink and Single Carrier Frequency Division Multiple Access (SC-FDMA) in the uplink. The initial releases of LTE rely upon the

Table 2.4 Objectives and features of the EVS codec

Enhanced quality and coding efficiency for narrowband (NB: 200–3,400 Hz) and wideband (WB: 50–7,000 Hz) speech
Enhanced quality by the introduction of super wideband (SWB: 50–14,000 Hz) speech
Enhanced quality for mixed content and music in conversational applications (e.g. in-call music)
Robustness to packet loss and delay jitter leading to optimized behavior in IP environments
Backward interoperability to AMR-WB by EVS modes supporting the AMR-WB codec format.
Source-controlled variable bit rate modes in addition to constant bit rate modes

AMR-NB/WB voice codecs for voice coding, but this is a stop gap effort while bodies work to develop a voice codec specifically for LTE for Enhanced Voice Services (EVS) [32]. Some objectives and features of the EVS voice codec for LTE are summarized in Table 2.4. Here we see that there is a desire to maintain interoperability with the AMR codecs while adding a superwideband capability and giving more attention to in-call music. As the EVS codec nears final characterization, there are some specific advances that will be widely deployed and used. First, new 5.9 kbps source controlled variable bit rate (VBR) modes for both narrowband and wideband speech that achieve the same quality as the AMR-NB/WB codec but at a lower average rate have been added to improve capacity. Further, there is better constant bit rate (CBR) coding of both WB and SWB music, and improved CBR coding of WB and SWB speech. Also included are optional full band and stereo modes for voice and music. Key design constraints are that codec delays up to 32 ms are allowed and a complexity up to twice that of AMR-WB, namely, 88 WMOPS.

High quality audio codecs for non-conversational services such as streaming, broadcasting, and multicasting also have been standardized earlier by 3GPP. These codecs are AMR-WB+ and aacPlus, but their high algorithmic delay restricts their importance for two-way conversational voice.

Another codec standardization effort had the goal of coding narrowband voice all the way up to fullband audio with the constraint of low delay. The Opus Audio Codec, standardized by the IETF, is designed for interactive voice and audio and has three modes [20, 21]: (a) A linear prediction based mode for low bit rate coding up to 8 kHz bandwidth, (b) A hybrid linear prediction and MDCT mode for fullband speech/audio at medium bit rates, and (c) an MDCT-only mode for very low latency coding of speech and audio. It has a wide range of bit rates from 6 kbps up to 510 kbps to support full band audio. Further, Opus has available frame sizes from 2.5 ms up to 60 ms and algorithmic delay in the range of 5–62.5 ms. Details of this codec can be found at [20, 21]. Speech quality tests indicate that the Opus codec produces excellent voice quality at medium rates of 20–40 kbps [23].

2.5 Emerging Topics

As the field of speech coding continues to evolve, new issues emerge, old challenges persist, and often old constraints are relaxed. It is clear from the prior discussions that extending the bandwidth covered by codecs is a high priority for essentially every standards activity. As stated earlier, wider bandwidths improve intelligibility, naturalness, and speaker identifiability. The advantages of incorporating wider bandwidths need to be elaborated further given the extraordinary efforts of the standards bodies to achieve ever-increasing bandwidth capabilities in codecs [33].

Consonants are key in the intelligibility of many words and phrases. The frequency content of consonants often occurs in the 4–14 kHz frequency range, which is partially encompassed by wideband speech, but much more so by the newer superwideband classification. There are other factors that arise in standard videoconferencing, VoIP, and even person-to-person calls that are addressed by wider bandwidths. In conference rooms and other such venues, there are natural reflections off walls and ceilings that can degrade communications, especially if a speaker moves away from a microphone or speakers are different distances from microphones. In person, a listener is able to use both ears, which greatly helps in alleviating misunderstandings, but for audio and video conference calls, often there is only one microphone and therefore only one channel being delivered to the other end. Experiments show that wider bandwidths aid greatly in reducing confusion and easing listener fatigue.

Another important point in this multinational business environment and with non-native English speakers routinely playing critical roles in organizations is accented speech. This point also holds for speakers within a country, such as the US, where there are quite different speech patterns. Speakers with accents will often have different pronunciations and different grammatical patterns. As a result, native listeners may not be able to correctly process sentences when there are different pronunciations because of the different grammatical structure. Increasing bandwidth provides considerable improvement in these situations.

Extending the lower end of the band is also of substantial value, since frequencies below 200 Hz add to listener comfort, warmth, and naturalness. It is thus very clear why the exceptional efforts to extend the bandwidths covered by codecs are being pursued.

Stereo audio is a new effort in communications applications. The capture of stereo, or more generally, multichannel audio, is simpler than it sounds, even for handheld devices. For example, there may be two microphones, one pointed toward the active speaker and the other outwardly to record the environment. There are many other microphone configurations that may be desirable as well [22]. As stereo audio capture and delivery becomes of interest, it is necessary to make decisions as to how to allocate bit rate; that is, if a choice must be made, is it preferable to send wider bandwidth speech/audio or stereo channels? Coupled to this question is how to evaluate the quality of the expanded bandwidths and additional multichannel audio when delivered to the user.

A recent addition to the perceptual quality evaluation area is the use of a nine point range for the MOS values [22, 32]. Unlike the five point scale, only the extreme values are given designations of Excellent (9) and Very Bad (1). It is shown that this scale allows the tests to be accomplished relatively quickly and that the various conditions are distinguished by the tests. Comparisons are given in [22] and [23], wherein the latter contains a performance evaluation of the Opus codec.

2.6 Conclusions and Future Research Directions

There are several clear trends in recent standardization efforts. First, single standard codecs that encompass the entire range of narrowband to fullband are highly desirable and the norm for the future. Second, while latency constraints have been relaxed for many applications, there is still a demand for lower latency codecs to be used in communications services. Third, increasing complexity is acceptable as long as the speech/audio quality is substantially improved. Fourth, there is a strong impetus to capture and code stereo channels for many applications, including handheld devices.

Another fact is also clear—the current standards still rely very heavily on the well-worn coding paradigms of code-excited linear prediction and transform/filter bank methods with noise masking. It is this fact that points to a great need for new research directions to try and identify new codec structures to continue the advance in speech/audio codec compression developments. Although standardization efforts have resulted in many new codec designs and the understanding of the basic structures, it is unlikely that given the time constraints and continuously competitive nature of codec standardization processes, these new research directions will be undertaken through the development of new standards.

Some suggested research directions are to incorporate increased adaptivity into the codec designs. For example, adapting the parameters of the perceptual weighting filters in CELP is one possible research direction. Another is to incorporate adaptive filter bank/transform structures such as adaptive band combining and adaptive band splitting. A third more difficult direction is to identify entirely new methods to incorporate perceptual constraints into codec structures.

It is hoped that the current chapter will motivate some of these new research efforts.

References

1. J.D. Gibson, Speech coding methods, standards, and applications. IEEE Circuits Syst. Magazine **5**, 30–49 (2005)
2. J.D. Gibson, T. Berger, T. Lookabaugh, D. Lindbergh, R.L. Baker, *Digital Compression for Multimedia: Principles and Standards* (Morgan-Kaufmann, San Francisco, 1998)

3. R. Cox, S.F. de Campos Neto, C. Lamblin, M.H. Sherif, ITU-T coders for wideband, superwideband, and fullband speech communication. IEEE Commun. Magazine **47**, 106–109 (2009)
4. ITU-T Recommendation P.862, Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs (2001)
5. ITU-T Recommendation P.862.2, Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs (2007)
6. ITU-T Recommendation P.863, Perceptual objective listening quality assessment (2011)
7. W.-Y. Chan, T.H. Falk, Machine assessment of speech communication quality, in *The Mobile Communications Handbook*, ed. by J.D. Gibson, 3rd edn. (CRC Press, BocaRaton, FL, 2012). Chapter 30
8. Advanced audio distribution profile (A2DP) specification version 1.2, Bluetooth SIG, Audio video WG, http://www.bluetooth.org/. April 2007
9. H.S. Malvar, *Signal Processing with Lapped Transforms* (Artech House, Norwood, 1992)
10. A.M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems* (Wiley, West Sussex, 2004)
11. J.H. Chen, A. Gersho, Adaptive postfiltering for quality enhancement of coded speech. IEEE Trans. Audio Process. **3**, 59–70 (1995)
12. S. Ragot et al., ITU-T G.729.1: An 8-32 kbit/s scalable coder interoperable with G.729 for wideband telephony and Voice over IP, in *Proceedings of ICASSP*, Honolulu, April 2007
13. ITU-T Recommendation G.722.1, Coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss (1999)
14. ITU-T Recommendation G.722.2, Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB) (2002)
15. ITU-T Rec. G.718, Frame error robust narrowband and wideband embedded variable bit-rate coding of speech and audio from 8-32 kbit/s (2008)
16. ITU-T Rec. 719, Low-complexity, full-band audio coding for high-quality, conversational applications, June 2008
17. S. Karapetkov, G.719: the first ITU-T standard for full-band audio. Polycom white paper, April 2009
18. http://www.speex.org/
19. S.V. Andersen, W.B. Kleijn, R. Hagen, J. Linden, M.N. Murthi, J. Skoglund, iLBC – a linear predictive coder with robustness to packet losses, in *Proceedings of the IEEE Speech Coding Workshop*, October 2002, pp 23–25
20. IETF Opus Interactive Audio Codec, http://opus-codec.org/ (2011)
21. RFC6716, Definition of the Opus Audio Codec, September 2012
22. A. Ramo, Voice quality evaluation of various codecs, in *ICASSP 2010*, Dallas, 14–19 March 2010
23. A. Ramo, H. Toukomaa, Voice quality characterization of the IETF Opus Codec, in *Proceedings of Interspeech 2011*, Florence (2011)
24. A. Ramo, H. Toukomaa, On comparing speech quality of various narrow- and wideband speech codecs, in *Proceeding of ISSPA*, Sydney (2005)
25. M. Bosi, R.E. Goldberg, *Introduction to Audio Coding and Standards* (Kluwer, Boston, 2003)
26. T. Painter, A. Spanias, Perceptual coding of digital audio. Proc. IEEE **88**, 451–512 (2000)
27. ITU-T Recommendation G.114, One-Way Transmission Time (2000)
28. ITU-T Rec. G.718 Amendment 2: New Annex B on superwideband scalable extension for ITU-T G.718 and corrections to main body fixed-point C-code and description text, March 2010
29. M. Neuendorf, P. Gournay, M. Multrus, J. Lecomte, B. Bessette, R. Geiger, S. Bayer, G. Fuchs, J. Hilpert, N. Rettelbach, F. Nagel, J. Robilliard, R. Salami, G. Schuller, R. Lefebvre, B. Grill, A novel scheme for low bitrate unified speech and audio coding-MPEG RM0, in *Audio Engineering Society,* Convention Paper 7713, May 2009

30. Y. Hiwasaki et al., G.711.1: a wideband extension to ITU-T G.711. *EUSIPCO 2008*, Lausanne, 25–29 August 2008
31. M. Xie, D. Lindbergh, P. Chu, ITU-T G.722.1 Annex C: a new low-complexity 14 kHz audio coding standard, in *Proceedings of ICASSP*, Toulouse, May 2006
32. K. Jarvinen, I. Bouazizi, L. Laaksonen, P. Ojala, A. Ramo, Media coding for the next generation mobile system LTE. Comput. Commun. **33**, 1916–1927 (2010)
33. J. Rodman, The effect of bandwidth on speech intelligibility. Polycom white paper, September 2006

Chapter 3
Scalable and Multi-Rate Speech Coding for Voice-over-Internet Protocol (VoIP) Networks

Tokunbo Ogunfunmi and Koji Seto

Abstract Communication by speech is still a very popular and effective means of transmitting information from one person to another. Speech signals form the basic method of human communication. The information communicated in this case is verbal or auditory information. The methods used for speech coding are very extensive and continuously evolving.

Speech Coding can be defined as the means by which the information-bearing speech signal is coded to remove redundancy thereby reducing transmission bandwidth requirements, improving storage efficiency, and making possible myriad other applications that rely on speech coding techniques.

The medium of speech transmission has also been changing over the years. Currently a large percentage of speech is communicated over channels using internet protocols. The voice-over-internet protocols (VoIP) channels present some challenges that have to be overcome in order to enable error-free, robust speech communication.

There are several advantages to use bit-streams that are multi-rate and scalable for time-varying VoIP channels. In this chapter, we present the methods for scalable, multi-rate speech coding for VoIP channels.

3.1 Introduction

Speech communication using the Voice over Internet Protocol (VoIP) [57, 59, 66, 78] is rapidly replacing the old but still ubiquitous circuit switched telephone service. However, speech packetized and transmitted through packet-switched

T. Ogunfunmi (✉) • K. Seto
Department of Electrical Engineering, Santa Clara University,
Santa Clara, CA 95053, USA
e-mail: togunfunmi@scu.edu

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_3

networks incurs numerous impairments including delay, jitter, packet loss and decoder clock offset, which degrade the quality of the speech. Advanced signal processing algorithms can combat these impairments and render the perceived quality of a VoIP conversation to be as good as that of the existing telephone system.

For example, the increased transport delay in VoIP networks renders the normally tolerable echoes to be more annoying. Jitter buffers are also essential to smooth-out the inevitable delay variations caused by the network routers. Packet loss can be a major source of impairments in long distance packet switched networks, and it is essential to use loss concealment algorithms to alleviate their effects in VoIP systems. While transmitter based Forward Error Correction (FEC) methods can be used to correct isolated packet losses, receiver based signal processing algorithms are generally preferred as they can work independent of the transmitter. The internet Low Bit Rate Coder (iLBC) is a recent speech coder that tries to mitigate some of the impairments caused by packet loss by incorporating some of these methods.

This chapter is organized as follows: We begin the next section with a brief introduction of Voice over IP Networks. Then in Sect. 3.3, we present an overview of Analysis-by-Synthesis Speech Coding. In Sect. 3.4, we discuss the Multi-rate Speech Coding. In Sect. 3.5, we present the Scalable Speech Coding and in Sect. 3.6, Packet Loss Robust Speech Coding is presented. Finally, we conclude and summarize in Sect. 3.7.

3.2 VoIP Networks

A typical voice call using the Public Switched Telephone Network (PSTN) proceeds as follows: Analog speech from the near end handset is first encoded at the originating exchange using the 64 kbps G.711 PCM standard; it is then transported through a series of 64 kbps Time Division Multiplexed (TDM) circuit switches to the terminating exchange where it is decoded back to the original analog waveform and sent to the far end handset. Since the TDM switches in the voice path have small frame buffers, and are all synchronized to a common reference clock by an overlay synchronization network, there is virtually no impairments to the switched voice samples. Therefore the PSTN is ideally suited for voice communications and the resulting speech quality is considered to be outstanding. However, it is not flexible for switching traffic with rates other than 64 kbps, and is also not efficient for transmitting bursty traffic. Moreover, it requires two separate networks: a circuit switched TDM network for the voice path and a packet switched Signaling System Number 7 (SS7) network for setting up and tearing down the connections ([1], etc.)

3.2.1 Overview of VoIP Networks

Compared to the PSTN, the packet switched internet protocol (IP) network natively supports variable bandwidth connections and uses the same network for both media and signaling communications. In a VoIP call, speech is first encoded at the transmitter using one of the voice encoding standards, such as G.711, G.726, or G.729. The encoded speech is then packetized using the Real Time Transport (RTP) protocol. After appending additional headers to complete the protocol stack, the packets are routed through the IP network. They are de-packetized and decoded back to analog voice at the receiver.

3.2.2 Robust Voice Communication

The packetization and routing processes used in a VoIP system necessitate speech buffers. The voice packets transported in such a network incur larger delays compared to the PSTN. They also arrive at the receiver unevenly spaced out in time due to the variation of the buffering delay in the path routers. This delay variation, known as jitter, must be smoothed out with a jitter buffer. Furthermore, in a typical IP network, the intermediate routers drop packets when they are overloaded due to network congestion. The ensuing gaps in the received packets have to be bridged with packet loss concealment algorithms. A further consequence of transporting voice through an IP network is that it is difficult to reproduce the original digitizing clock at the receiving end, and therefore the frequency of the (independent) playout clock generally differs from that of the sampling clock. This leads to underflow or overflow situations at the receive buffer, as the voice samples are written into it at the original voice digitizing rate but read out at a different rate. Such a clock skew problem has to be corrected using silence interval manipulation or speech waveform compression/expansion techniques.

Satisfactory communication of voice using the packet IP network therefore demands that the effect of the above-mentioned impairments such as delay, jitter, etc. be mitigated using proper signal processing algorithms. Reference [33] deals with this subject.

3.2.3 Packet Loss Concealment (PLC)

The decoder will apply Packet Loss Concealment (PLC) techniques when packets are lost or don't arrive in time for playback. A PLC unit is designed for the decoder to recognize when a packet has been lost and masks the effect of losing a packer or having a considerable delay in its arrival.

We give an example of a PLC unit that can be used with any codec using Linear Prediction (LP). The traditional PLC unit is generally used only at the decoder, and therefore the PLC unit does not affect interoperability between implementations. Other PLC implementations may therefore be used.

The example Packet Loss Concealment unit addresses the following cases:

1. *Current and previous frames are received correctly.*
   The decoder saves the state information (LP filter coefficients) for each sub-frame of the current frame and entire decoded excitation signal in case the following frame is lost.
2. *Current frame is not received.*
   If the frame is not received, the frame substitution is based on a pitch-synchronous repetition of the excitation signal, which is filtered by the last LP filter of the previous frame. The previous frame's information is stored in the decoder state structure.

   The decoder uses the stored information from the previous frame to perform a correlation analysis and determine the pitch lag and voicing level (the degree to which the previous frame's excitation was a voiced or roughly periodic signal). The excitation signal from the previous frame is used to create a new excitation signal for current frame to maintain the pitch from the previous frame.

   For a better sounding substituted frame, a random excitation is mixed with the new pitch periodic excitation, and the relative use of the two components is computed from the correlation measure (voicing level).

   Next, the signal goes through a LP filter to produce a speech output that makes up for the lost packet/frame.

   For several consecutive lost frames, the packet loss concealment continues in a similar manner. The correlation measure of the last frame received is still used along with the same pitch value. The LP filters of the last frame received are also used again. The energy of the substituted excitation for consecutive lost frames is decreased, leading to a dampened excitation, and therefore to dampened speech.
3. *Current frame is received, but previous frame is lost.*
   In this case, the current frame is not used for the actual output of the decoder. In order to avoid an audible discontinuity between the current frame and the frame that was generated to compensate for the previous packet loss, the decoder will perform a correlation analysis between the excitation signal of both frames (current and previous one) to detect the best phase match. Then a simple overlap-add procedure is performed to merge the previous excitation smoothly into the current frames' excitation.

The exact implementation of the packet loss concealment does not influence interoperability of the codec.

3.3 Analysis-by-Synthesis Speech Coding

The Linear Predictive Coding (LPC) model of speech generation is based on the acoustic model of speech generation using the vocal tract excitation.

Now, we briefly introduce the concept of Code-Excited Linear Predictive (CELP) [77] speech coders. It is the basis of many of the modern speech coders using the so-called analysis-by-synthesis method [62–65].

3.3.1 Analysis-by-Synthesis Principles

The powerful method of speech coding using analysis-by-synthesis ensures that the best possible excitation is chosen for a segment of speech. Many speech coders are based on this method.

In Fig. 3.1, a basic block diagram of an analysis-by-synthesis encoder is shown. A decoder is embedded in the encoder. This is a closed-loop system. The parameters are extracted by the encoding, and then they are decoded and used to synthesize the speech. The synthetic speech is compared with the original speech and the error is minimized (in a closed loop) to further choose the best parameters in the encoding. The measures of the minimization include MSE, etc.

3.3.2 CELP-Based Coders

We briefly introduce the CELP methods for low bit-rate speech coding. First we discuss the limitations of LPC as a way to introduce necessity for CELP coders.

The quality of speech generated by the LPC model depends on the accuracy of the model. The LPC model is quite simplistic in assuming that each frame of

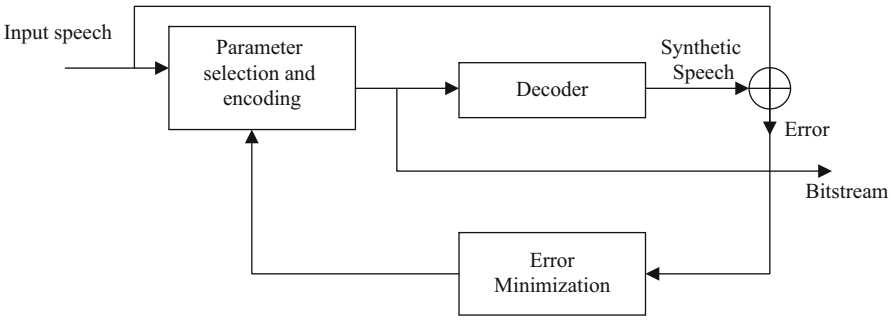

Fig. 3.1 Block diagram of an encoder based on the analysis by synthesis principle (incorporates a decoder)

speech can be classified as voiced or unvoiced. In reality, there are some brief regions of transitions between voiced and unvoiced and vice-versa that the LPC model incorrectly classifies. This can lead to artifacts in the generated speech which can be annoying.

The fixed choice of two excitations: white noise or periodic impulses is not truly representative of the real speech generation models especially for voiced speech. In addition, the nature of the periodic pulses used is not truly periodic, nor are they truly impulses. This leads to synthetic speech that is not truly natural-sounding.

Natural-ness can be added to the synthetic speech by preserving some of the phase information which is not typically preserved during the LPC process, especially for voiced frames. Unvoiced frame phase information can be neglected. This is important even though the human ear is relatively insensitive to phase information.

The spectrum generated by exciting a synthesis filter with periodic impulses (as required for LPC modeling of generation of voiced frames) is one that is distorted. This is due to a violation of the requirement that the AR model be excited by a flat-spectrum excitation (which is true of white noise). Use of a periodic impulse train for excitation however, leads to a distorted spectrum. This is more noticeable for low-pitch period voiced speech like that of women and children. For such speech, LPC-based synthetic speech is not very good.

In order to alleviate some of these problems with LPC, the CELP has been developed. The CELP uses a long-term and a short-term synthesis filter to avoid voiced/unvoiced frame decision. It also uses phase information by combining the high quality potential of waveform coding with the compression efficiency of parametric model-based vocoders.

A vector of random noise is used as an excitation instead of white noise/impulse train. The Multi-pulse LPC excitation idea has been extended to vector excitation. However, the excitation vectors are stored at both transmitter and receiver. Only the index of the excitation is transmitted. This leads to large reduction in bits transmitted. Vector quantization is required for vector excitation. The CELP speech coders are a result of this.

The codebook contains the list of possible vectors determined from minimization of the overall distortion measure such as weighted mean square of the quantization error. The codebook can be fixed or adaptive.

The basic block diagrams of a CELP-based encoder and decoder are shown in Figs. 3.2 and 3.3 respectively. Notice that there are two parts of the synthesis filter: the long-term prediction (LTP) pitch synthesis filter and the short-term prediction (STP) formant synthesis filter. Also, the error in the synthesized speech is perceptually weighted and then minimized. Then it is used to determine the proper index for getting the excitation from the Fixed codebook. The codebook contains a vector of possible excitations for the synthesis filters. A vector excitation is better than having to make a hard choice between impulse train and white noise as possible excitations in the LPC model.

At the decoder, the index is used to determine the proper excitation needed from the Fixed Codebook. The excitation is multiplied by the gain and then used to excite

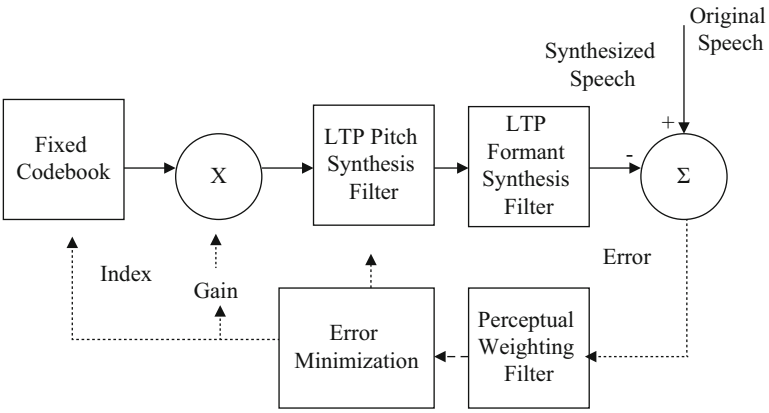

Fig. 3.2 Basic structure of encoder of CELP coders

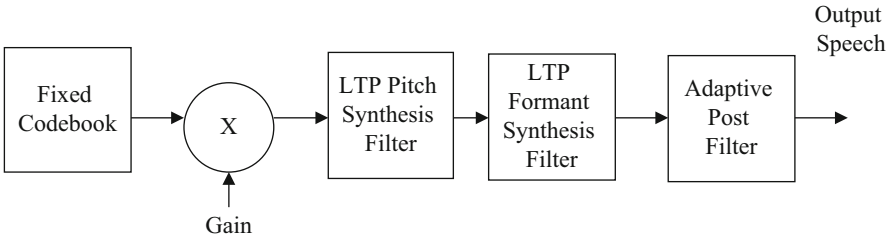

Fig. 3.3 Basic structure of decoder of CELP coders

the LTP pitch synthesis filter and the output is then used to excite the STP formant synthesis filter. An adaptive post filter is used to smooth the output speech.

The STP is the normal LPC filter that models the envelope of the speech spectrum. Its transfer function is given by

$$\frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

The STP coefficients are generally obtained once per frame using the autocorrelation equations and transformed to *line spectral frequencies* (LSF) for quantization and transmission.

The LTP models the fine structure (pitch) of the speech spectrum. Its transfer function can be written as

$$\frac{1}{P(z)} = \frac{1}{1 - \beta z^{-L}}$$

where β denotes the pitch gain and L denotes the pitch lag. These parameters are determined at sub-frame intervals using a combination of open and closed loop techniques.

The fixed codebook generates the excitation vector that is filtered by the LTP and STP to generate the synthesized speech signal. The index of the codebook and the gain is obtained so that a perceptually weighted error between the original and synthesized speech signals is minimized.

3.3.2.1 Perceptual Error Weighting

The LTP and codebook parameters are selected to minimize the mean square of the perceptually weighted error sequence. The perceptual weighting filter is given by

$$W(z) = \frac{A(z)}{A\,(z/\gamma)} = \frac{1 - \sum_{k=1}^{p} a_k z^{-k}}{1 - \sum_{k=1}^{p} a_k \gamma^k z^{-k}}, \quad 0 < \gamma < 1$$

The weighting filter de-emphasizes the frequency regions corresponding to the formants as determined by the STP filter, thereby allocating more noise to the formant regions, where they are masked by the speech energy, and less noise to the subjectively disturbing regions close to the frequency nulls. The amount of de-emphasis is controlled by the parameter γ ($\gamma = 0.75$ in ITU-T G729A standard). A more general weighting filter of the form

$$W(z) = \frac{A\,(z/\gamma_1)}{A\,(z/\gamma_2)}$$

is employed in certain CELP coders (e.g., ITU-T G.729 standard).

3.3.2.2 Pitch Estimation

Determining the accurate pitch period of the speech signal is a difficult task. This task is often broken up into two stages to reduce computational complexity. First an open-loop search is performed, over the whole range of possible values of pitch period to obtain a coarse estimate. The estimate is then refined using a closed loop (analysis-by-synthesis) technique. Fractional pitch delay estimates are generally required to synthesize good quality speech.

The **open-loop pitch analysis** is normally done once per frame. The method of coarse pitch estimation basically consists of calculating the autocorrelation function

of the weighted speech signal $s_w(n)$ and choosing the delay L that maximizes it. One problem with this approach is that multiple pitch periods within the range of values of L might occur if the pitch period is small. In this case there is a possibility that the first peak in the autocorrelation is missed and a sub-multiple of the pitch chosen, thereby generating a lower-pitched voice signal. To avoid this pitch multiples problem, the peak of the autocorrelation is estimated in several lag ranges (three ranges in ITU-T G729 and G729A standards), and smaller pitch periods are favored in the selection process with proper weighting of the normalized autocorrelation values.

Closed-Loop Pitch Search (Adaptive Codebook Search): Significant improvement in voice quality can be achieved when the LTP parameters are optimized inside the analysis-by-synthesis loop. We first assume that the codebook output is zero. The pitch delay L is then selected as the delay (in the neighborhood of the open loop estimate) that minimizes the mean square of the perceptually weighted error. The optimum pitch gain value is usually obtained by a simple codebook search.

If the pitch period L is greater than the length of the sub-frame over which the codebook search is performed, the contribution to the synthetic speech for this sub-frame is only a function of the excitation sequence that was used in the last sub-frame, which is stored in the LTP buffer, and is not a function of the current choice of the fixed codebook excitation sequence. With this interpretation, the pitch synthesis filter can be viewed as an adaptive codebook that is in *parallel* with the fixed codebook.

3.4 Multi-Rate Speech Coding

3.4.1 Basic Principles

The objective of speech coding is to represent speech signals in a format that is suitable for digital communication. Traditionally, the focus in the codec design has been to minimize bit rate subject to some quality requirements. However, in practice, the design of speech codecs is primarily governed by application needs and constraints.

In traditional public switched telephone network (PSTN), a fixed bit rate was used for each communication direction regardless of factors such as short-term characteristics of speech signals, transmission channel conditions, or network load. In contrast, a bit rate can be varied by a function of these factors in modern communication networks in order to improve performance of speech codecs.

The PSTN was designed to have very low error rates, whereas bit errors and packet loss are inherent in modern communication infrastructures. Bit errors are common in wireless networks and are generally handled by channel coding. Packet loss occurs in IP networks and is typically concealed by a speech codec.

On the other hand, a variable rate speech codec [67] can be employed to adapt its bit rate to current channel conditions in order to mitigate bit errors or packet loss. Variable rate speech codecs can generally be divided into two main categories [2–4, 58]:

- **Source-controlled** variable rate codec, where the data rate is determined by the short-term characteristics of speech signals.
- **Network-controlled** variable rate codec, where the data rate is determined by an external control signal generated by the network in response to channel conditions.

In source-controlled coding, the codec dynamically allocates bits depending on the short-term characteristics of speech signals, and therefore, the average bit rate is typically less than the bit rate of the fixed rate codec to achieve a given level of speech quality. Note that the source-controlled coding scheme can be combined with the network-controlled coding scheme. The Telecommunications Industry Association (TIA) selected a variable bit-rate (VBR) codec called QCELP [5], which was developed by Qualcomm, to increase the capacity of a code division multiple access (CDMA)-based digital cellular system, and IS-96 algorithm [6] was standardized in 1994. It was later replaced with the enhanced variable rate codec (EVRC), standardized by the TIA as IS-127 [7].

In network-controlled coding, the codec responds to an external control signal to switch the data rate to one of a predetermined set of possible rates. The network-controlled coding scheme can be viewed as multi-mode variable rate coding scheme or multi-rate coding scheme, where multiple modes of speech coding are defined with each mode having a different fixed bit rate.

A special case of multi-rate coding called bit-rate scalable speech coding is of particular interest and is explained in detail in the next section [Sect. 3.5]. In bit-rate scalable speech coding, a bit-stream has a layered structure with a core bit-stream and enhancement bit-streams. Enhancement layers are added to a core layer to improve speech quality. During transmission of scalable bit-streams, the bit rate can be adaptively reduced by truncating the enhancement layer bit-streams according to network conditions. However, a scalable speech codec generally has lower performance than a multi-rate codec with each bit-rate mode independently optimized for the highest speech quality.

In 1999, the Adaptive Multi-Rate (AMR) speech codec has been standardised by European Telecommunication Standards Institute (ETSI) [8–10]. The Third Generation Partnership Project (3GPP) adopted the AMR codec as the mandatory speech codec for the third generation WCDMA system in April 1999 [11]. The wideband version of the AMR codec is referred to as Adaptive Multirate Wideband (AMR-WB) codec [12] and was standardized in 2001 [13]. The AMR and AMR-WB are supported in all Universal Mobile Telecommunications System (UMTS) and Long Term Evolution (LTE) terminals [14]. The next sub-section provides the brief descriptions of the AMR speech codec, which is one of the most widely used speech codec in wireless telephony.

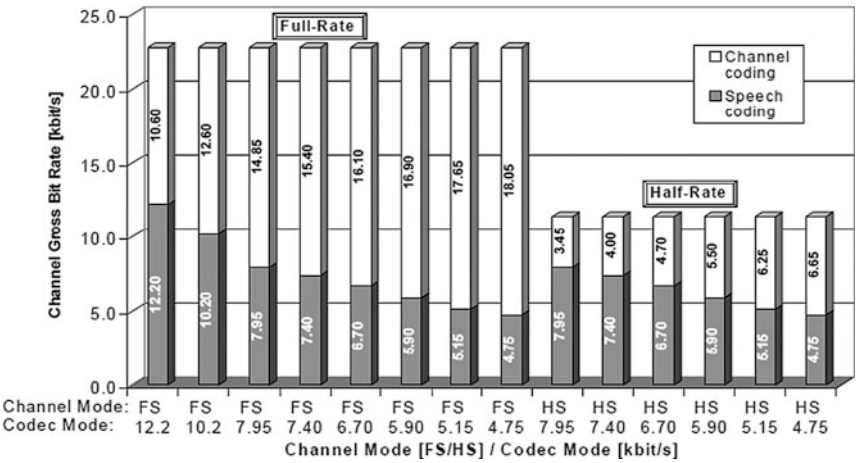

Fig. 3.4 Bit-rate trade-off between speech and channel coding [8]

3.4.2 Adaptive Multi-Rate (AMR) Codec

An overview of the AMR speech codec [10] used for GSM systems is provided here. The AMR codec is based on the Algebraic CELP (ACELP) coding scheme [15]. The AMR has two channel modes: full-rate (FR) and half-rate (HR). The FR mode provides a gross bit rate of 22.8 kbps, whereas the HR mode provides a gross bit rate of 11.4 kbps. The AMR is capable of operating at eight different codec modes (bit rates): 4.75, 5.15, 5.9, 6.7, 7.4, 7.95, 10.2 and 12.2 kbps. Note that The 12.2 and 7.4 kbps modes are equivalent to GSM enhanced full-rate (EFR) [16] and IS-136 EFR [17], respectively. There are a total of 14 different combinations of operational choices as shown in Fig. 3.4. A gross bit rate consists of the speech coding bit rate and channel coding bit rate and each mode has a different distribution of a gross bit rate between speech and channel coding as illustrated in Fig. 3.4, which results in a different level of error protection. The codec selects the optimum channel mode and codec mode to deliver the best combination of speech quality and system capacity according to the current radio channel and traffic load conditions. The higher bit-rate mode generally offers better speech quality with lower error protection. In contrast, the lower bit-rate mode offers lower speech quality with higher error protection. The channel quality selection and the selection of the optimum channel and codec modes are performed by link adaptation process [18]. The codec mode can be changed every 40 ms.

3.5 Scalable Speech Coding

3.5.1 Basic Principles

Scalable speech coding is a special case of multi-rate coding with a bitstream structured into layers which consists of a core bitstream and enhancement bitsrteams as already explained in Sect. 3.4.1. When a scalable speech codec is used, the encoder can operate at the highest bit rate, but some enhancement layers could be discarded at any point of communication systems by simply truncating the bitstream to reduce the bit rate. In contrast to a multi-rate codec, a feedback of channel conditions and re-encoding are not required for a scalable codec to reduce the bit rate. Therefore, a layered bitstream offers higher flexibility and easier adaptation to sudden change of network conditions, which can be exploited to reduce packet loss rates. In fact, enhancement layers can be used to add various types of functionality to a core layer, such as speech quality improvement (also called signal-to-noise ratio (SNR) scalability), bandwidth extension or mono to stereo extension (number of channels extension) [19].

There are two other advantages [20] for employing scalable speech codecs. First, scalable coding is a possible solution to cope with the heterogeneity and variability in communication systems. In fact, the telephone industry has been experiencing a transition from the PSTN to an all IP network. Currently, links having different capacities and terminals with various capabilities may coexist. A transmission path may include both wireless links and fixed links with different capacities. Using a scalable coding approach, users can receive different quality versions of the same speech according to their individually available resources and supported capabilities without the need of feedback.

Secondly, the coexistence of the PSTN and IP networks with a mixture of wireless and fixed links means that transcoding at gateways is inevitable. In this situation, the bitstream scalability can be employed to ensure interoperability with different network infrastructures. In addition, a scalable extension of a widely used core coder is a very attractive solution to develop and deploy a new enhanced codec while maintaining interoperability and backwards compatibility with existing infrastructure and terminals.

In the following section, two examples of the state-of-the-art standardized scalable wideband codecs, ITU-T G.729.1 and ITU-T G.718, are introduced.

3.5.2 Standardized Scalable Speech Codecs

In this section, two scalable speech codecs which have been standardized within the “International Telecommunication Union–Telecommunication Standardization Sector” (ITU-T) are described.

3.5.2.1 ITU-T G.729.1

ITU-T G.729.1 [21, 25] is a scalable wideband codec. It offers a 12-layer scalable (embedded) bitstream structure with bit rates between 8 and 32 kbit/s, and is interoperable with widely deployed G.729 [22–24].

The codec operates on 20 ms frames (called superframes), although the core layer and the first enhancement layer at 12 kb/s operate on 10 ms frames similar to G.729. A major novelty was that G.729.1 provides bit rate and bandwidth scalability at the same time. Bit rates at 8 and 12 kb/s are narrowband. The wideband rates range from 14 to 32 kb/s at 2 kb/s intervals.

G.729.1 is the first speech codec with a scalable structure built as an extension of an already existing standard. It offers full backward bitstream interoperability at 8 kb/s with the much used G.729 standard in voice over IP (VoIP) infrastructures. G.729.1 is one of the best for wideband speech quality, and its quality is preserved regardless of the access modes and device capabilities thanks to strong robustness to IP packet losses.

3.5.2.1.1 Encoder and Decoder

The G.729.1 encoder and decoder are illustrated in Fig. 3.5a, b, respectively. By default, both input and output signals are sampled at 16 kHz, and the encoder operates at the maximal bit-rate of 32 kbit/s. The input $s_{WB}(n')$ is decomposed into

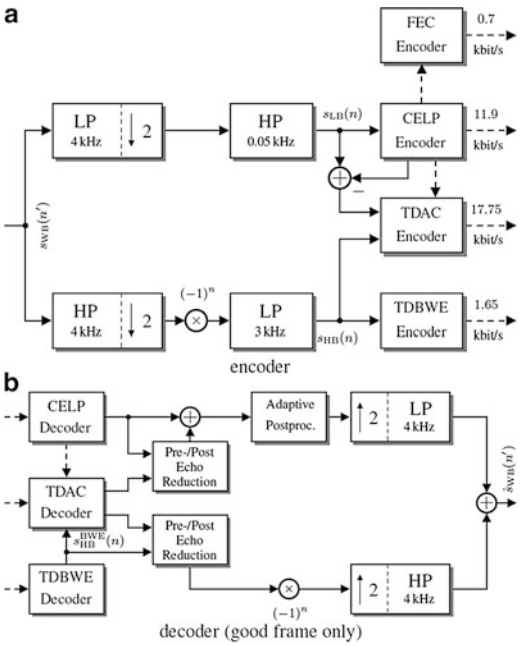

Fig. 3.5 Block diagrams of the G.729.1 encoder and decoder [25] (**a**) encoder and (**b**) decoder (good frame only)

two subbands using a 64-coefficient analysis quadrature mirror filterbank (QMF). The lower band is pre-processed by an elliptic high-pass filter (HPF) with 50 Hz cutoff and encoded by a cascade (or two-stage) CELP coder. The higher band is spectrally folded, pre-processed by an elliptic low-pass filter (LPF) with 3 kHz cutoff, and encoded by parametric time-domain bandwidth extension (TDBWE). The lowerband CELP difference signal and the higher-band signal $s_{HB}(n)$ are jointly encoded by the so-called time-domain aliasing cancellation (TDAC) encoder which is a transform-based coder. To improve the resilience and recovery of the decoder in case of frame erasures, parameters useful for frame erasure concealment (FEC) are transmitted by the FEC encoder based on available lower-band information.

The decoder operates in an embedded manner depending on the received bit-rate. At 8 and 12 kbit/s the CELP decoder reconstructs a lower-band signal (50–4,000 Hz) which is then post-filtered in a way similar to G.729; the result is upsampled to 16 kHz using the QMF synthesis filterbank. At 14 kbit/s, the TDBWE decoder reconstructs a higher-band signal $s^{BWE}(n)$ which is combined with the 12 kbit/s synthesis in order to extend the output bandwidth to 50–7,000 Hz. From 16 to 32 kbit/s, the TDAC decoder reconstructs both a lower-band difference signal and a higher-band signal, which are then post-processed (shaped in time domain) to mitigate pre/post echo artifacts due to transform coding. The modified TDAC lowerband signal is added to the CELP output, while the modified TDAC higher-band synthesis is used instead of the TDBWE output to improve quality for the whole frequency range.

3.5.2.1.2 Robust Encoding Approach

The FEC procedure is derived from the FEC/recovery part of the 3GPP2 VMR-WB speech coder (in generic full-rate encoding type) [26]. At the encoder, 14 bits per superframe are used to send supplementary information, which improves FEC and the recovery of the decoder after frame erasures. The FEC parameters consist of signal classification information (2 bits), phase information (7 bits) and energy (5 bits). They are distributed in Layers 2, 3 and 4 respectively, so at to minimize the impact of bits "stolen" to the cascade CELP, TDBWE and TDAC coding stages.

The FEC follows a split-band approach: in lower band the LPC excitation is reconstructed and filtered by the estimated LPC synthesis filter; in the higher band, the decoder is supplied with the previously received TDBWE time and frequency envelope parameters—the TDBWE mean-time envelope is attenuated by 3 dB after each erasure.

3.5.2.2 ITU-T G.718

The ITU-T G.718 [28, 69, 71] is an embedded codec comprising five layers; referred to as L1 (core layer) through L5 (the highest extension layer). The lower two layers are based on Code-excited Linear Pre-diction (CELP) technology. The core layer, derived from the VMR-WB speech coding standard [27], comprises several

Table 3.1 Layer structure for default operation [28]

Layer	Bit-rate (kbit/s)	Technique	Internal sampling rate
L1	8	Classification-based core layer	12.8 kHz
L2	+4	CELP enhancement layer	12.8 kHz
L3*	+4	FEC / MDCT	12.8 / 16 kHz
L4*	+8	MDCT	16 kHz
L5*	+8	MDCT	16 kHz

*Not used for NB input-output

coding modes optimized for different input signals. The coding error from L1 is encoded with L2, consisting of a modified adaptive codebook and an additional fixed algebraic codebook. The error from L2 is further coded by higher layers (L3–L5) in a transform domain using the modified discrete cosine transform (MDCT). Side information is sent in L3 to enhance frame erasure concealment (FEC). The layering structure is summarized in Table 3.1 for the default operation of the codec.

The encoder can accept wideband (WB) or narrowband (NB) signals sampled at either 16 or 8 kHz, respectively. Similarly, the decoder output can be WB or NB, too. Input signals sampled at 16 kHz, but with bandwidth limited to NB, are detected and coding modes optimized for NB inputs are used in this case. The WB rendering is provided for, in all layers. The NB rendering is implemented only for L1 and L2. The input signal is processed using 20 ms frames. Independently of the input signal sampling rate, the L1 and L2 internal sampling frequency is at 12.8 kHz.

The codec delay depends upon the sampling rate of the input and output. For WB input and WB output, the overall algorithmic delay is 42.875 ms. It consists of one 20 ms frame, 1.875 ms delay of input and output re-sampling filters, 10 ms for the encoder look-ahead, 1 ms of post-filtering delay, and 10 ms at the decoder to allow for the overlap-add operation of higher-layer transform coding. For NB input and NB output, the 10 ms decoder delay is used to improve the codec performance for music signals, and in presence of frame errors. The overall algorithmic delay for NB input and NB output is 43.875 ms; 2 ms for the input re-sampling filter, 10 ms for the encoder look-ahead, 1.875 ms for the output re-sampling filter, and 10 ms decoder delay. Note that the 10 ms decoder delay can be avoided for L1 and L2, provided that the decoder is prevented from switching to higher bit rates. In this case the overall delay for WB signals is 32.875 ms and for NB signals 33.875 ms.

3.5.2.2.1 Codec Structure

The structural block diagram of the encoder for WB inputs is shown in Fig. 3.6. From the figure it can be seen that while the lower two layers are applied to a pre-emphasized signal sampled at 12.8 kHz as in [12], the upper three layers operate at the input signal sampling rate of 16 kHz.

To optimize the speech quality at 8 kb/s, the CELP core layer of G.718 uses signal classification and four distinct coding modes tailored to different classes of input signal: voiced coding (VC), unvoiced coding (UC), transition coding (TC), and

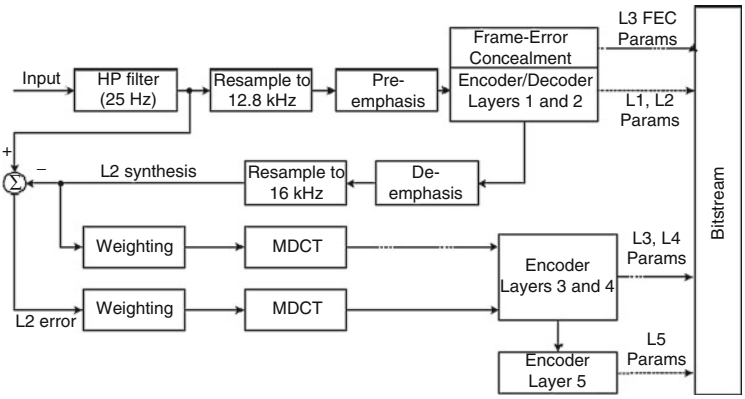

Fig. 3.6 Block diagram of the G.718 encoder [28]

generic coding (GC). VC and GC employ the algebraic CELP (ACELP) technology as in [12]. The VC mode is used for encoding voiced speech frames without significant variation in periodicity, so relatively fewer bits can be allocated to the adaptive codebook and more bits to the algebraic codebook than is the case for GC. The UC mode is used for encoding stable unvoiced speech frames. As no periodicity can generally be observed in unvoiced speech, the adaptive codebook is not used, and the excitation is composed of two vectors selected from a small Gaussian codebook using a fast search. The TC mode is used for encoding frames following transitions from unvoiced to voiced speech. As frames containing such transitions are most sensitive to frame erasures due to error propagation into subsequent frames, the TC mode has been designed to severely limit any prediction from past frames. This is done in particular by replacing the adaptive codebook in the beginning of a TC mode frame with a small fixed codebook of eight stored glottal shapes [29]. Finally, the GC mode is used for encoding frames not classified otherwise. Independent of the first layer coding mode, the coding error from the core layer is then encoded using an additional algebraic fixed codebook in the second layer.

The encoding of L3 and L4 is different for speech-dominant content and music-dominant content, with discrimination based on the coding efficiency of the CELP model. For speech dominant content, the whole MDCT spectrum is coded at the fourth layer level using scalable algebraic vector quantization with bits covering the lower frequencies sent in L3 and the remaining bits in L4. In the case of music input, the second layer synthesis is first attenuated to reduce noisiness generated by CELP. The L3 MDCT coefficients are then quantized only in a selected band [30], where the band selection is based on the energy of the MDCT coefficients. In L4, the entire 7 kHz bandwidth is coded using an unconstrained pulse position vector quantizer known as factorial pulse coding (FPC) [31]. FPC is also used systematically in L5, independent of the input signal.

3.5.2.2.2 Robust Encoding Approach

The codec has been designed with emphasis on frame erasure (FE) conditions with several techniques limiting frame error propagation. The TC mode has already been described in the previous subsection. In addition, the LP filter quantization in some frames and the excitation gain coding do not use interframe prediction. To further enhance the speech quality in FE conditions, side information is sent in L3 to better control the decoded signal energy and periodicity in case of lost frames. The side information consists of class information for all coding modes. Previous frame spectral envelope information is also transmitted if the TC mode is used in the core layer. For other core layer coding modes, phase information and the pitch-synchronous energy of the synthesized signal are transmitted. The G.718 concealment strategy is different for different classes of speech signal (voiced, unvoiced, transitions) [32] and depends principally on the signal class of the last correctly received frame.

3.6 Packet-Loss Robust Speech Coding

Advances and wide acceptance of voice over Internet protocol (VoIP) [33] have been driving the evolution of telephony technologies in recent years. Voice communication over IP networks has gained popularity and may become the dominant service for overall telephony including the wireless telephony in the near future [33, 34]. According to Federal Communications Commission (FCC) [35], the technological advisory council (TAC) made a number of draft recommendations to the FCC, which include exploring end dates to the PSTN. On the other hand, the emergence of VoIP has posed new challenges to development of speech codec. The key issue of transporting real-time voice packet over IP networks is the lack of guarantee for reasonable speech quality due to packet delay or loss.

The characteristics of IP network channel is constantly changing. Especially, packet traffic on public Internet can be unpredictable and its channel is expected to produce much higher packet delay or loss rate than managed networks. Therefore, voice communication over public Internet is less reliable. Reliability of VoIP can be increased by controlling IP networks. The IP multimedia subsystem (IMS) is a network functional architecture for multimedia service delivery and suited for controlling the multimedia traffic by utilizing quality of service (QoS). It uses a VoIP implementation based on session initiation protocol (SIP), and runs over the standard IP. By relying on managed networks using IMS, packet loss rate is reduced and bursty loss pattern of IP networks becomes less severe although the packet loss is still the main cause of performance degradation and a codec that is robust to packet loss is required. Therefore, the functionality of speech codec that allows its bit rate to adapt to the current available channel capacity is of significant importance because the efficient channel usage is maintained by adjusting the congestion of packet traffic. The RTP control protocol (RTCP) is used along with the real-time transport protocol (RTP) to provide feedback on the quality of speech transmission

for VoIP applications. However, RTCP is not always enabled and feedback is slow. Therefore, the adaptation to the current channel condition without the need of feedback is the attractive feature of VoIP application due to the requirement of short time delay for real-time communication. Bit-stream scalability is a promising technique that makes it possible to adjust the bit rate to the desired value by truncating the bit stream at any point of a communication system. Low packet delay can be maintained by adjusting the bit rate of voice traffic. Note that the benefits of scalability are most enjoyed by the codec used for public Internet. The jitter buffer management (JBM) can also be used to mitigate the effect of delay jitter and is achieved by buffering incoming packets at the receiver and delaying their playout so that most of the packets are received before their scheduled playout times.

Recently, scalable speech coding techniques [20] have become the subject of intense research, and the need for scalable speech coding has been clearly recognized by the industry, resulting in new standardization activities. Indeed, bit-stream scalability facilitates the deployment of new codecs that are built as embedded extensions of widely deployed codecs. Most of the recent scalable speech codecs including the wideband scalable speech codec such as ITU-T G.729.1 or G.718 depend on CELP coding technique for core layer and low bit rate operations. The CELP technique utilizes the long-term prediction (LTP) across the frame boundaries and therefore causes error propagation in the case of packet loss and need to transmit redundant information in order to mitigate the problem. Some of the simple solutions were proposed in [36], which requires significant increase in bit rate and delay. Recent approach was introduced in [37] to reduce the error propagation after lost frames by replacing the long-term prediction with a glottal-shape codebook in the subframe containing the first glottal impulse in a given frame, and utilized in G.718. Another approach which depends on low bit-rate redundancy frames and an LTP scaling parameter can be found in the recent codec called Opus [38].

The internet low bit-rate codec (iLBC) [39, 40] employs the frame-independent coding and therefore inherently possesses high robustness to packet loss. When packets are lost, the effect of speech quality degradation is limited without depending on transmission of redundant information. Note that this benefit of high robustness to packet loss comes at the expense of a high bit rate. Due to its inherent robustness to packet loss, iLBC quickly became a popular choice of speech codecs for VoIP applications and was adopted by Skype and Google Talk. However, the lack of flexibility in terms of data rates and its relatively high operational bit rate compared to CELP-based codecs have overshadowed the advantage of iLBC. In order to overcome those shortcomings, the rate-flexible solutions for iLBC were introduced in [41, 42]. The scalable structure was integrated to iLBC in [43, 44].

A wideband codec provides significant improvement over a narrowband codec in terms of speech intelligibility and naturalness. The transition from narrowband to wideband telephony is in progress. During this transition, ensuring interoperability and backwards compatibility with existing infrastructure and terminals is essential. One of the attractive solutions is to extend the capabilities of existing narrowband codecs to provide wideband coding functionality by using the bandwidth extension

technique. The bandwidth scalable structure can be used to extend bandwidth by adding the enhancement layer to the core layer. Enhancement layers can also provide speech quality or SNR improvement. Therefore, the wideband support was added to the narrowband speech codec based on the iLBC in [45, 46] by employing the bandwidth extension.

We describe the details of the original iLBC coding scheme in Sect. 3.6.1. The scalable multi-rate speech codec based on the iLBC coding scheme is presented in Sect. 3.6.2. Where the narrowband and wideband codecs are separately described and the performance evaluation results are also provided.

3.6.1 Internet Low Bitrate Codec (iLBC)

The IP environment can lead to degradation in speech quality due to lost frames, which occurs in connection with lost or delayed IP packets.

The iLBC is a speech codec designed for robust voice communication over IP networks. It was designed for narrow band speech signals sampled at 8 kHz. The algorithm uses a block-independent linear-predictive coding (LPC) algorithm and has support for two basic frame/block lengths: 20 ms at 15.2 kbit/s and 30 ms at 13.33 kbit/s. When the codec operates at block lengths of 20 ms, it produces 304 bits per block, which is packetized as in [39, 47] (or fits in 38 bytes). Similarly, for block lengths of 30 ms it produces 400 bits per block, which is packetized (or fits in 50 bytes). The two modes for the different frame sizes operate in a very similar way.

The algorithm results in a speech coding system with a controlled response to packet losses similar to what is known from Pulse Code Modulation (PCM) with packet loss concealment (PLC), such as the ITU-T G.711 standard, which operates at a fixed bit rate of 64 kbit/s. At the same time, the algorithm enables fixed bit rate coding with a quality-versus-bit rate tradeoff better than most other algorithms.

The iLBC coder is suitable for real time communications such as telephony and videoconferencing, streaming audio, archival, and messaging. It is commonly used for VoIP applications such as Skype, Yahoo Messenger and Google Talk among others. Cable Television Laboratories (CableLabs) has adopted iLBC as a mandatory PacketCable audio codec standard for VoIP over Cable applications.

The structure of the iLBC was developed by a company called Global IP Solutions (GIPS) formerly Global IP Sound (acquired by Google Inc. in 2011). It uses the Real-time Transport Protocol (RTP) payload format. The ideas behind the iLBC were also presented at the IEEE Speech Coding Workshop in 2002 [39, 47].

This codec overcomes dependency of Code Excited Linear Prediction (CELP) codec (e.g. G.729, G.723.1, GSM-EFR and 3GPP-AMR) on previous samples. For packet-based networks using any of these CELP codecs, packet loss will affect the quality of the reconstructed signal as part of the historical information may be lost making it difficult to recover the original signal.

The computational complexity [56] of the iLBC is in the same range as the ITU-T G.729/A codec. It has the same quality as the ITU-T G.729E in clean channel (no

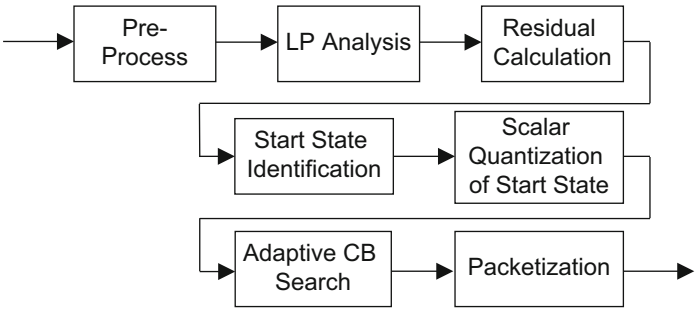

Fig. 3.7 Block diagram of the iLBC encoder

packet loss) conditions but its quality exceeds that of the other narrow-band codecs including G.723.1 and G.728 under packet loss conditions.

Figure 3.7 shows the block diagram of iLBC speech encoder. The basic framework of iLBC is based on linear prediction (LP) model and block based coding of LP residual signal using an adaptive codebook (CB) as are used on CELP-based codecs. The main difference from CELP-based codecs is that the long-term predictive coding is exploited without introducing inter-frame dependencies. Therefore, the propagation of errors across frames is avoided when packets are lost, which makes the iLBC robust to packet loss. This frame independence is achieved by applying the adaptive CB both forward and backward in time, starting from the start state. The start state captures pitch information in voiced speech and accurate noise-like information in unvoiced speech, and enables the operation of the adaptive CB without depending on the history of the LP residual signal. The adaptive CB search is repeated three times for refinement.

The benefit of using the start state comes at the expense of a large number of bits required to represent it accurately for each frame. The start state occupies 43.5 and 56.25 % of encoded bits for 30 ms frame mode and 20 ms frame mode, respectively.

3.6.2 Scalable Multi-Rate Speech Codec

The scalable multi-rate speech codec based on the iLBC coding scheme which optimized for narrowband input is presented first. The scalable wideband multi-rate codec is introduced subsequently.

3.6.2.1 Narrowband Codec

The scalable narrowband speech codec based on the iLBC was developed in two steps: the addition of rate flexibility to the iLBC and the addition of scalability to the multi-rate codec based on the iLBC. These two types of codecs are described separately in the following subsections. The performance evaluation of the developed narrowband codec is provided subsequently.

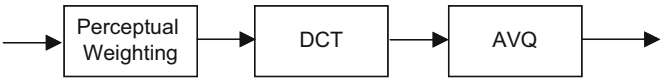

Fig. 3.8 Block diagram of DCT-based start state encoder

3.6.2.1.1 Multi-Rate Codec Based on the iLBC

Since the start state contains the important information as explained in the Sect. 3.6.1, the encoding process should maintain its waveform as accurately as possible. The original iLBC uses 3-bit scalar quantizer. However, a time domain waveform coding is not flexible in terms of the bit rate reduction. A frequency domain coding technique has potential for reducing the bit rate because of the nature of the start state. The discrete cosine transform (DCT) is used since it has a strong energy compaction property, a fast transform algorithm is available and the start state is completely independent for each frame.

Figure 3.8 shows the block diagram of the start state encoder using the DCT, which replaces the block for scalar quantization of the start state in Fig. 3.7. The N samples of the start state $x_0, \ldots, x_k$ are processed by perceptual weighting filter

$$W(z) = \frac{1}{\widehat{A}\left(\frac{z}{\gamma_s}\right)}$$

where $\widehat{A}(z)$ is a LP analysis filter and the filter $W(z)$ models the short-term frequency masking curve. The parameter γ_s is used to adjust the degree in which the perceptual weighting is applied. Note that the start state is in the residual domain and weighting the start state with $W(z)$ is equivalent to employing a perceptual weighting filter $\widehat{A}(z)/\widehat{A}(z/\gamma_s)$ in speech signal domain as used in CELP technique. The filter $W(z)$ is initialized to zero in each frame. Note also that N is one of the parameters in the proposed codec ranging from 40 to 80 whereas N for the original iLBC is 57 and 58 for the 20 and 30 ms frame, respectively. The weighted start state samples are transformed into the DCT coefficients $X_0, \ldots, X_k$ by one-dimensional DCT according to

$$X_k = w_k \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right) \quad k = 0, \ldots, N-1$$

where

$$w_k = \begin{cases} 1/\sqrt{N} & k = 0 \\ \sqrt{2/N} & 1 \le k \le N-1 \end{cases}$$

The DCT coefficients are quantized by the scalable algebraic vector quantization (AVQ) which is specified in G.718 [28, 75] and implemented by forming 8-dimensional vectors and using multi-rate lattice vector quantizer [60, 76]. To remain within the total bit budget, DCT coefficients are divided by a global gain prior to quantization.

The multi-rate functionality is obtained by allocating different number of bits to the AVQ. When a small number of bits are available to use for the AVQ, those bits are allocated to only a limited number of sub-bands.

The low bit rate operation is achieved by decreasing the number of available bits for the AVQ, which leads to rapid degradation of speech quality. Two schemes were already introduced in [48] to improve performance at low bit rates. One of the schemes is to reduce the number of adaptive CB stages and reallocate bits from one or two of the adaptive CB refinement stages to start state encoding. Another scheme is to reduce the length of the start state. These schemes sacrifice speech quality at high bit rates in order to achieve good speech quality at low bit rates.

Longer start state samples can capture more information and have better frequency resolutions. Therefore, high speech quality can be maintained at lower bit rates by reducing the number of the adaptive CB stages and reallocating a part of bits to the start state encoding. Especially when the length of the start state is 80, extra 34 bits can be saved since the first target of sub-frame to be encoded using the adaptive CB in the original iLBC is completely included in the start state.

3.6.2.1.2 Scalable Multi-Rate Codec Based on the iLBC

The scalable multi-rate speech codec using the iLBC is presented in this sub-section. The core layer coding error is encoded by employing the modified DCT (MDCT) and the AVQ. Figure 3.9 shows the block diagram of our proposed scalable multi-rate iLBC encoder. The input speech signal is encoded by multi-rate iLBC encoder. The bit-stream produced constitutes the core layer portion of the scalable bit-stream. The decoded speech signal is obtained during the iLBC encoding process. The multi-rate iLBC coding error is computed by subtracting the decoded speech signal from the original speech signal and processed by perceptual weighting filter $W_e = \widehat{A}(z/\gamma_e)$ where $\widehat{A}(z)$ is a LP analysis filter the parameter γ_e is used to adjust the degree in which the perceptual weighting is applied. This weighting filter is used

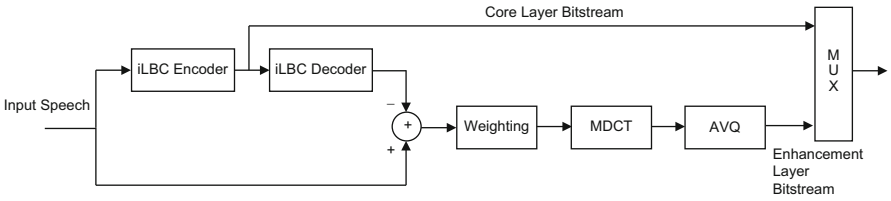

Fig. 3.9 Block diagram of scalable multi-rate iLBC encoder

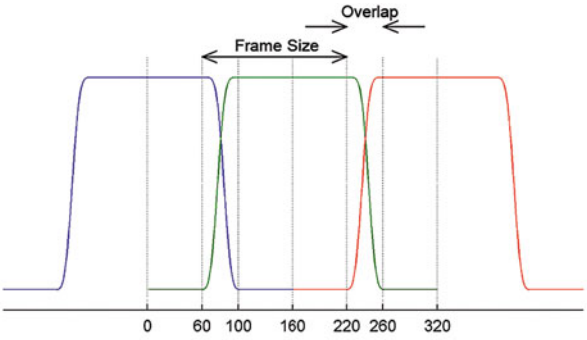

Fig. 3.10 Window function with reduced overlap for 20 ms frame mode. KBD window is used for overlap region

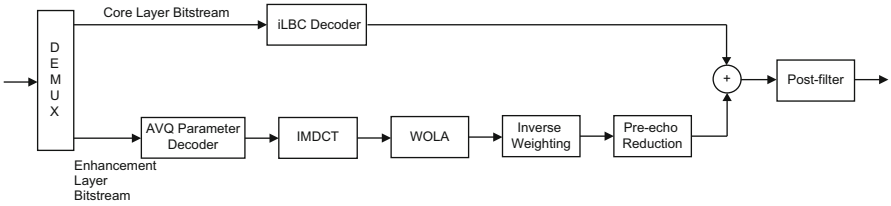

Fig. 3.11 Block diagram of scalable multi-rate iLBC decoder

to flatten MDCT coefficients as employed in G.729.1 and G.718. The weighted error signal is windowed and transformed into MDCT coefficients. Figure 3.10 shows the power-complementary window for 20 ms frame mode. For the overlap region, the Kaiser-Bessel derived (KBD) window is employed. To reduce the delay, the overlap is only 40 samples which correspond to 5 ms while the window size is 320 samples which is twice the frame size as shown in Fig. 3.10. The effective overlap can be reduced by padding zeros on each side and the perfect reconstruction is still achieved as long as the window function satisfies the Princen–Bradley condition [49]. The overall algorithmic delay for 20 and 30 ms frame mode is therefore 25 and 35 ms, respectively. The resulting MDCT coefficients are quantized using the AVQ and the enhancement layer bit-stream is produced.

The block diagram of scalable multi-rate iLBC decoder is shown in Fig. 3.11. The AVQ parameters of enhancement layer are decoded, transformed into time domain signal using inverse MDCT (IMDCT), and the weighted overlap-and-add (WOLA) synthesis is performed to obtain the perceptually weighted error signal. The weighted error signal is inverse-weighted and processed by the pre-echo reduction module which performs the same algorithm used in [28] to obtain the decoded error signal. The decoded speech signal of the core layer is combined with the error signal decoded from the enhancement layer. The enhanced speech signal

is passed through the post-filter to produce the output speech signal. The post-filter used in G.729.1 was modified to be incorporated in the proposed scalable iLBC by employing open-loop pitch estimation for the integer part of the pitch delay.

The post-processing unit used to enhance LP residual signals in the original iLBC was modified to be employed without adding any delay and is employed in the core layer of the decoder to achieve higher speech quality. Note that the post-processing unit needs to be included in the decoding process in the encoder as well. The post-processing unit in the original iLBC introduces 5 and 10 ms delay for 20 and 30 ms frame case, respectively in order to achieve high performance and is employed in the multi-rate iLBC when enhancement layer is not used. Therefore, the overall delay for the multi-rate iLBC without enhancement layer is 40 ms when the frame length is 30 ms, whereas the overall delay for the scalable multi-rate iLBC is 35 ms. When the frame length is 20 ms, the overall delay is 25 ms for the proposed codec with or without enhancement layer.

3.6.2.1.3 Performance Evaluation

In order to evaluate the quality of speech produced by the scalable narrowband multi-rate codec based on the iLBC, the objective tests based on PESQ algorithm [40, 50] were performed. The speech samples utilized for performance evaluation are from database in Annex B of ITU-T P.501 [51] pre-published in January 2012. The source speech was down-sampled to 8 kHz and its speech level was equalized to −26 dBov using the ITU-T Software Tool Library [52]. The modified-IRS filter and any mask were not used because the target VoIP applications includes soft phones.

Figure 3.12 shows the MOS-LQO scores of speech codecs as a function of bit rates. The best performance curves for core layer and core layer plus enhancement layer of the proposed scalable multi-rate iLBC are shown in Fig. 3.12. It also compares the proposed codec with G.718, G.729.1 and also with AMR. The bit allocations for the proposed codec operated at 12.2 kbps are provided in Table 3.2. Note that proposed scalable codec outperforms non-scalable (core layer only) configuration at bit rates higher than 10 kbps. The benefit of enhancement layer can be clearly seen. Interestingly, the performance of proposed codec is comparable to all the codecs under clean channel conditions despite the fact that the proposed codec is based on frame-independent coding. Especially, the proposed codec achieves the same MOS-LQO scores as G.729.1 at 12 kbps and almost the same scores as AMR at around 5 kbps. Even the performance of G.718 is close to that of the proposed codec at 12 kbps.

Figure 3.13 shows the performance comparisons of the proposed scalable codec with G.718, G.729.1, AMR, and the original iLBC operated at around 12.2 kbps under lossy channel conditions. The Gilbert Elliot channel model [53] is employed using the ITU-T Software Tool Library [52] to simulate the bursty packet loss such as the behavior of IP networks. The correlation of the packet loss is set to 0.2. Note that the bit rate of original iLBC is 13.3 kbps which is more than 1 kbps higher than other codecs. When packet loss rates are higher than 3 %, the performance

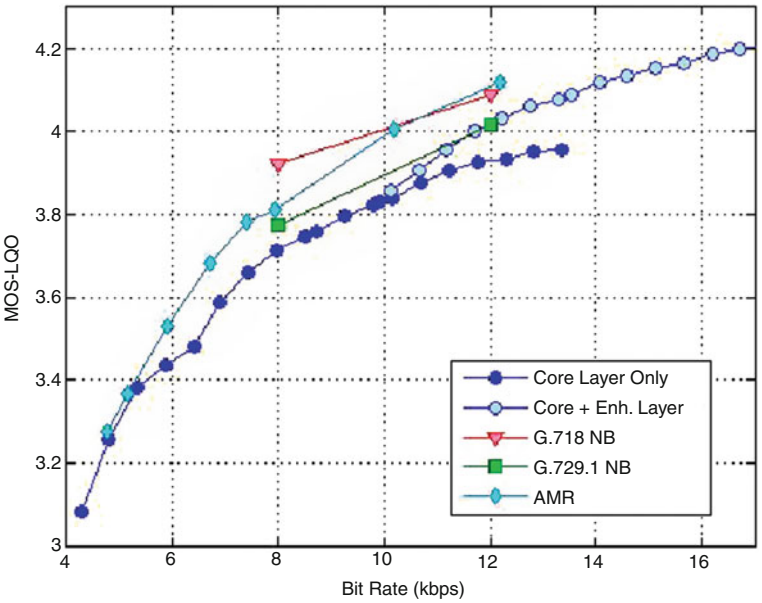

Fig. 3.12 Performance comparisons of the scalable multi-rate codec based on the iLBC with G.718, G.729.1, and AMR under clean channel condition

Table 3.2 Bit allocation when operating at 12.2 kbps

Parameter	Bits
LSF	40
Position of start state	3
DCT global gain for start state	7
DCT spectral parameters for start state	144
Adaptive CB index	63
Adaptive CB gain	36
MDCT spectral parameters for enhancement Layer	72
Empty frame indicator	1
Total	366

of AMR is worst and G.729.1 shows the second worst performance. Although the performance of the proposed codec, original iLBC, and G.718 are close to each other, the original iLBC achieves the highest scores at packet loss rates higher than 10 % and the proposed codec outperforms G.718 at the loss rates higher than 15 % and G.718 has the highest MOS-LQO score at 3 % packet loss rate. However when we take into account the effect of the PLC algorithm, the different conclusion can be drawn. The PLC algorithm used in G.718 is optimized for itself whereas the PLC scheme used in the proposed codec and the original iLBC is the informative only algorithm described in the original iLBC document which can be used in residual

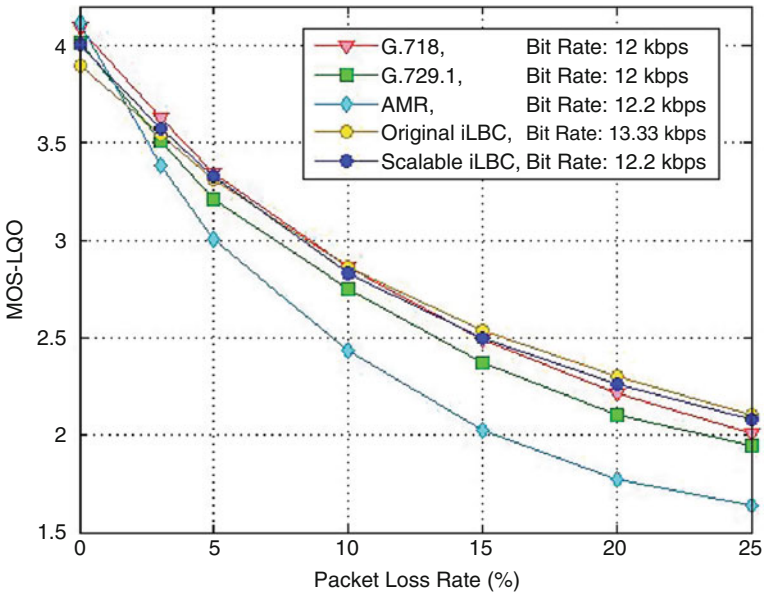

Fig. 3.13 Performance comparisons of the scalable multi-rate codec based on the iLBC with G.718, G.729.1, AMR, and original iLBC under lossy channel conditions when operated at around 12.2 kbps

domain for any codec. If the PLC algorithm is optimized for the proposed codec, higher robustness to packet loss can be expected. Therefore, the proposed codec is expected to outperform G.718 at most of the loss rates. Please note that the original iLBC outperformed the proposed codec at high packet-loss rates because the PLC algorithm used works only for the core layer and the original iLBC operates at the higher bit rate.

3.6.2.2 Wideband Codec

In this subsection, a scalable wideband speech codec based on the iLBC is presented.

3.6.2.2.1 Codec Structure

The basic codec structure is a scalable wideband extension of the multi-rate codec based on the iLBC. Figure 3.14 shows the block diagram of the encoder. The encoder operates on 20 ms input frames. The wideband input signal is sampled at 16 kHz and split into two sub-bands using a Quadrature Mirror Filter (QMF) analysis filter bank. The lower-band signal is first processed by a high-pass filter

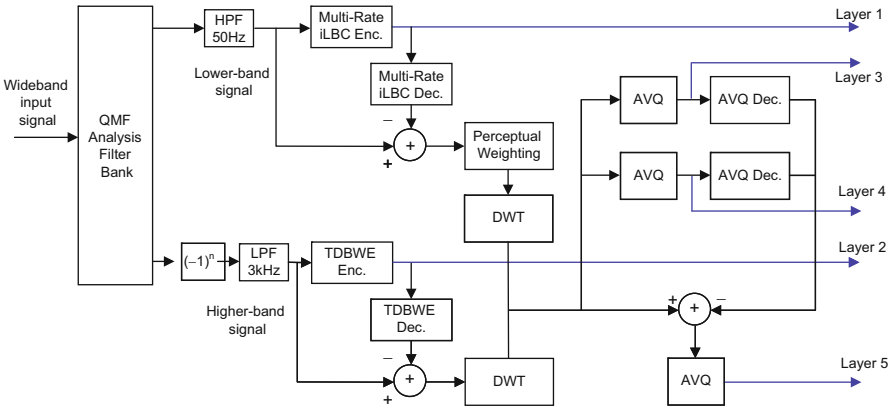

Fig. 3.14 Block diagram of the encoder

with 50 Hz cut-off frequency and encoded by the multi-rate iLBC using 80 start state samples, which generates the core layer (Layer 1) bitstream. The multi-rate iLBC coding error is computed by subtracting the decoded speech signal from the original speech signal and processed by perceptual weighting filter. The weighted error signal is transformed into DWT coefficients.

The higher-band signal is first spectrally folded and processed by low-pass filter with 3 kHz cut-off frequency. The low-pass filtered signal is encoded by the TDBWE and Layer 2 bitstream is generated. The DWT is applied to the coding error by the TDBWE and the DWT coefficients are obtained.

The resulting two sets of DWT coefficients cover whole wideband of input signal. Those DWT coefficients are divided into two parts at either 1 or 2 kHz and each part is separately quantized using the scalable AVQ and Layer 3 and Layer 4 bitstreams are produced. In order to further improve performance, the quantization errors from Layers 3 and 4 are encoded by the scalable AVQ, which generates Layer 5 bitstream.

The bitstream produced by the encoder is scalable. The enhancement layers can be truncated during transmission and speech signal is still decoded with decreased quality.

Note that the TDBWE algorithm is the same as the one employed in G.729.1 except that a fixed random sequence is used for the TDBWE residual signal in the decoder so that the TDBWE coding error can be used to improve performance.

The block diagram of the decoder is illustrated in Fig. 3.15. Each layer of bitstreams is decoded by the respective decoders and the decoded signals are added to generate the lower-band signal and the higher-band signal. After post-filtering the decoded lower-band signal and spectrally folding the decoded higher-band signal, both resulting signals are delay-adjusted and combined using a QMF synthesis filter bank.

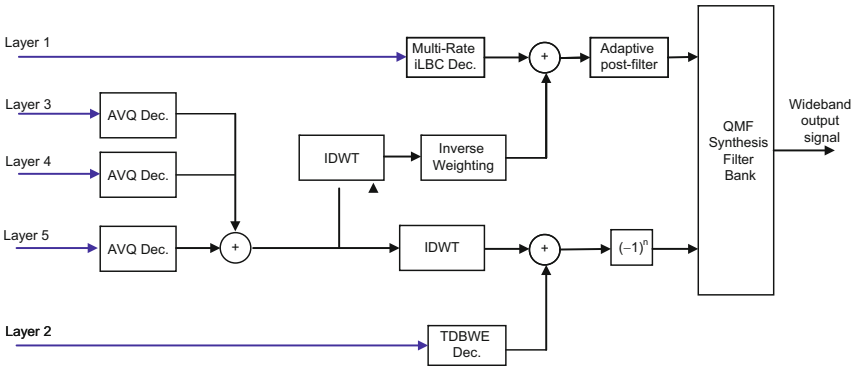

Fig. 3.15 Block diagram of the decoder

The DWT can be used to better capture localized waveforms in time domain than the Fourier-based transforms such as MDCT. The proposed codec utilizes the DWT to encode the lower- and higher-band coding error which is more likely to consist of highly non-stationary signals. Therefore, better performance can be expected by replacing the MDCT with the DWT.

In the proposed codec, we used the orthogonal Daubechies wavelet [54] with order 4, and three levels of decomposition for lower-band signals and one level of decomposition for higher-band signals. The wavelet coefficients are divided into six sub-bands when lower- and higher-band wavelet coefficients are combined. The delay from DWT is 6.125 ms, which is only 1.125 ms longer than the delay of 5 ms caused by the MDCT with reduced-overlap window used in Fig. 3.10.

The enhancement unit [73] in linear prediction (LP) residual domain used in the original iLBC decoder is employed in the multi-rate iLBC, which causes 5 ms delay. Therefore, the overall algorithmic delay is 40.0625 ms, which consists of 20 ms for input frame, 10 ms for the enhancement unit in the encoder and the decoder, 6.125 ms for the DWT, and 3.9375 ms for the QMF analysis-synthesis filterbank.

In order to improve performance under lossy channel conditions, the proposed codec employs the similar PLC algorithm to the one used in G.729.1. In the lower band, the G.729.1 PLC algorithm is slightly modified so that it works for the proposed codec in LP residual domain. In particular, some parameters which are not available in the decoder of the proposed codec are estimated. In the higher band, whereas the basic function of the TDBWE decoder is to shape an artificially generated excitation signal according to received time and frequency envelopes, only the TDBWE mean-time envelope and the frequency envelopes of the previous frame are used to shape an excitation signal when the frame is not received. The energy of the concealed signal is gradually decreased for the consecutive lost frames.

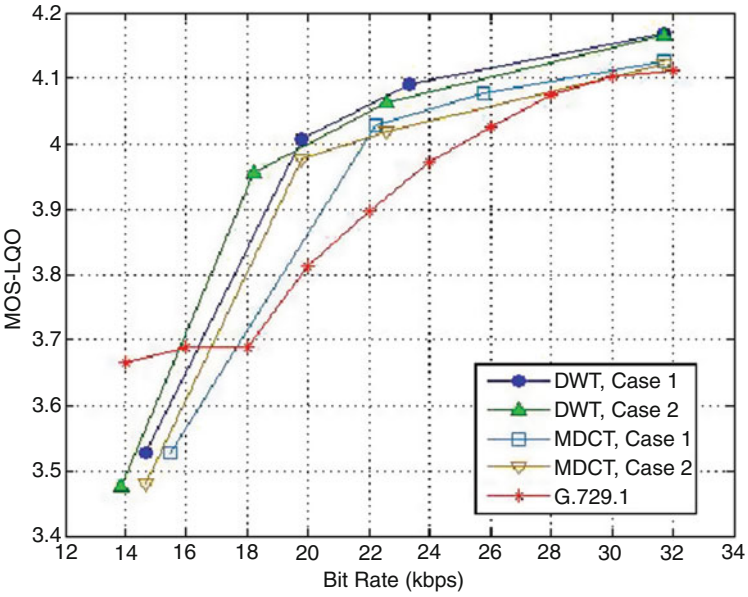

Fig. 3.16 Performance comparisons of the proposed wideband codec with G.729.1

3.6.2.2.2 Performance Evaluation

In order to evaluate the quality of speech produced by the scalable wideband multi-rate codec based on the iLBC, the objective tests based on PESQ algorithm were performed. The speech samples utilized for performance evaluation are from database in Annex B of ITU-T P.501 [51]. The source speech was down-sampled to 16 kHz and its speech level was equalized to −26 dBov. The modified-IRS filter and any mask were not used because the target VoIP applications includes soft phones.

Figure 3.16 shows performance comparisons of the proposed wideband codec using the DWT and the MDCT with G.729.1. The performances of two different parameter settings indicated as Case 1 and 2 are included for both the DWT and the MDCT in Fig. 3.16. It is clear that the performances of the proposed codec using either the DWT or the MDCT are higher than that of G.729.1 at most of the bit rates except at the low bit rates. The sudden drop of the codec performance at low bit rates is because a certain level of core-layer performance needs to be maintained. It is possible for the proposed codec to achieve similar performance to G.729.1 at low bit rates if lower performance is allowed for the core layer. It is also obvious that the DWT works better than the MDCT.

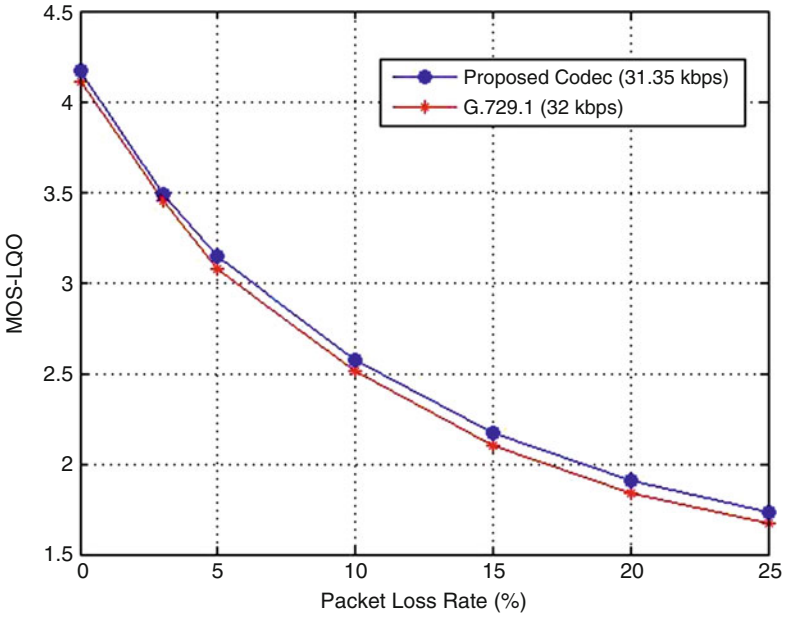

Fig. 3.17 Performance comparison of the proposed wideband codec with G.729.1 under lossy channel conditions

Note that the proposed codec and G.729.1 have a similar structure as both codecs are scalable extension of the narrowband codec to allow backward compatibility. It is remarkable that the proposed codec features the frame-independent coding scheme to achieve high robustness to packet loss and still performs better than G.729.1, which is CELP-based codec, at most of high bit rates.

Figure 3.17 shows the performance comparison of the proposed codec operated at 31.35 kbps with G.729.1 operated at 32 kbps under lossy channel conditions where the MOS-LQO scores are plotted as a function of packet loss rates. The Gilbert Elliot channel model is employed to simulate the bursty packet loss such as the behavior of IP networks. The correlation of the packet loss is set to 0.2. We can observe that the performance of the proposed codec is higher than that of G.729.1 at all packet loss rates. Both codecs employ the same type of PLC algorithm; however, the proposed codec needs to estimate some parameters which are not available at the decoder, including all the frame erasure concealment parameters. Therefore, the performance improvement due to the PLC algorithm under lossy channel condition should be much higher for G.729.1. We can say that the frame-independent coding in iLBC contributed to high robustness to packet loss. If the PLC algorithm is optimized for the iLBC-based core layer, better performance can be expected for the proposed codec.

3.7 Conclusions

Currently a large percentage of speech is communicated over channels using internet protocols. Most of the challenges presented by voice-over-internet protocols (VoIP) have been overcome in order to enable error-free, robust speech communication. However, robustness and scalabaility are two of the most important challenges remaining.

In this chapter, we presented a thorough discussion of scalable and multi-rate speech coding for Voice-over-IP networks. We discussed Voice-over-IP networks, Analysis by Synthesis coding methods, Multi-rate and Scalable methods of speech coding. We presented (as examples) new narrowband and wideband scalable and multirate speech codecs designed based on the iLBC, a robust codec deployed on the Voice-over-iP network channels.

References

1. J. Skoglund et al., Voice over IP: speech transmission over packet networks, in *Handbook of Speech Processing*, ed. by J. Benesty, M.M. Sondhi, Y. Huang (Berlin, Springer, 2009). Chap. 15
2. A. Gersho, E. Paksoy, An overview of variable rate speech coding for cellular networks, in *Proc. of the Int. Conf. On Selected Topics in Wireless Communications*, Vancouver (1992)
3. A. Gersho, E. Paksoy, Variable rate speech coding for cellular networks, in *Speech and Audio Coding for Wireless and Network Applications*, ed. by B.S. Atal, V. Cuperman, A. Gersho (Kluwer Academic, Norwell, 1993), pp. 77–84
4. V. Cuperman, P. Lupini, Variable rate speech coding, in *Modern Methods of Speech Processing*, ed. by R.P. Ramachandran, R.J. Mammone (Kluwer Academic, Norwell, 1995), pp. 101–120
5. W. Gardner, P. Jacobs, C. Lee, QCELP: a variable rate speech coder for CDMA digital cellular, in *Speech and Audio Coding for Wireless and Network Applications*, ed. by B.S. Atal, V. Cuperman, A. Gersho (Kluwer Academic, Norwell, 1993), pp. 85–92
6. TIA, Speech service option standard for wideband spread spectrum systems—TIA/EIA/IS-96 (1994)
7. TIA, Enhanced variable rate codec, speech service option 3 for wideband spread spectrum digital systems—TIA/EIA/IS-127 (1997)
8. K. Järvinen, Standardization of the adaptive multi-rate codec, in *Proceedings of European Signal Processing Conference (EUSIPCO)*, Tampere (2000)
9. E. Ekudden, R. Hagen, I. Johansson, J. Svedberg, The AMR speech coder, in *Proc. IEEE Workshop on speech coding*, Porvoo (1999), pp. 117–119
10. ETSI, Digital cellular telecommunications system (Phase 2+); Adaptive multi-rate (AMR) speech transcoding, GSM 06.90, version 7.2.1, Release (1998)
11. ETSI, Universal mobile telecommunications system (UMTS); Mandatory speech codec speech processing functions AMR speech codec; Transcoding Functions, 3GPP TS 26.090 Version 3.1.0, Release (1999)
12. B. Bessette et al., The adaptive multirate wideband speech codec (AMR-WB). *IEEE Trans. Speech Audio Process.* **10**, 620–636 (2002)
13. ETSI, Adaptive multi-rate – wideband (AMR-WB) speech codec; Transcoding functions, 3GPP TS 26.190 (2001)
14. K. Järvinen et al., Media coding for the next generation mobile system LTE. *Elsevier Comput. Commun.* **33**(16), 1916–1927 (2010)

15. C. Laflamme, J-P. Adoul, R. Salami, S. Morisette, P. Mabilleau, 16 kbps wideband speech coding technique based on algebraic CELP, in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, Toronto (1991), pp. 13–16
16. K. Järvinen et al., GSM enhanced full rate speech codec, in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, Munich (1997), pp. 771–774
17. T. Honkanen et al., Enhanced full rate speech codec for IS-136 digital cellular system, in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, Munich (1997), pp. 731–734
18. S. Bruhn, P. Blöcher, K. Hellwig, J. Sjöberg, Concepts and solutions for link adaptation and inband signaling for the GSM AMR speech coding standard, in *IEEE Vehicular Technology Conference* (1999)
19. Y. Hiwasaki, T. Mori, H. Ohmuro, J. Ikedo, D. Tokumoto, A. Kataoka, Scalable speech coding technology for high-quality ubiquitous communications. *NTT Tech. Rev.* **2**(3), 53–58 (2004)
20. B. Geiser et al., Embedded speech coding: from G.711 to G.729.1, in *Advances in Digital Speech Transmission*, ed. by R. Martin, U. Heute, C. Antweiler (Wiley, Chichester, 2008), pp. 201–247. Chap. 8
21. ITU-T Rec. G.729.1, An 8–32 kbit/s Scalable Wideband Coder Bitstream Interoperable with G.729, International Telecommunication Union (ITU) (2006)
22. ITU-T Rec. G.726, Adaptive Differential Pulse Code Modulation (ADPCM) of Voice Frequencies, International Telecommunication Union (ITU) (1990)
23. ITU-T Rec. G.728, Coding of Speech at 16 kbit/s Using Low-Delay Code-Excited Linear Prediction (LD-CELP), International Telecommunication Union (ITU) (1992)
24. ITU-T Rec. G.729, Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP), International Telecommunication Union (ITU) (1996)
25. S. Ragot, B. Kovesi, R. Trilling, D. Virette, N. Duc, D. Massaloux, S. Proust, B. Geiser, M. Gartner, S. Schandl, H. Taddei, Y. Gao, E. Shlomot, H. Ehara, K. Yoshida, T. Vaillancourt, R. Salami, M.S. Lee, D.Y. Kim. ITU-T G.729.1: an 8–32 kb/s scalable coder interoperable with G.729 for wideband telephony and voice over IP, in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing* (2007), pp. 529–532
26. TIA, Source-controlled variable-rate multimode wideband speech codec (VMR-WB)—3GPP2 C.S0052-0 (2004)
27. M. Jelínek, R. Salami, Wideband speech coding advances in VMR-WB standard. *IEEE Trans. Audio Speech Lang. Process.***15**(4), 1167–1179 (2007)
28. T. Vaillancourt et al., ITU-T G.EV-VBR: a Robust 8–32 kb/s scalable coder for error prone telecommunications channels, in *Proceedings of the Eusipco*, Lausanne, Switzerland (2008)
29. V. Eksler, M. Jelínek, Transition coding for source controlled CELP codecs, in *Proc. IEEE ICASSP*, Las Vegas (2008), pp. 4001–4004
30. M. Oshikiri et al., An 8–32 kb/s scalable wideband coder extended with MDCT-based bandwidth extension on top of a 6.8 kb/s narrowband CELP coder, in *Proceedings of Interspeech*, Antwerp (2007), pp.1701–1704
31. U. Mittal, J.P. Ashley, E. Cruz-Zeno. Low complexity factorial pulse coding of MDCT coefficients using approximation of combinatorial functions, in *Proceedings of IEEE ICASSP*, Honolulu, vol. 1 (2007), pp. 289–292
32. T. Vaillancourt et al., Efficient frame erasure concealment in predictive speech codecs using glottal pulse resynchronisation, in *Proceedings of IEEE ICASSP*, Honolulu, vol. 4 (2007) pp. 1113–1116
33. T. Ogunfunmi, M.J. Narasimha, Speech over VoIP networks: advanced signal processing and system implementation. *IEEE Circuits Syst.* Magazine **12**(2), 35–55 (2012)
34. FCC, http://transition.fcc.gov/oet/tac/TACMarch2011mtgfullpresentation.pdf, Meeting presentation of the Technological Advisory Council (2011a)
35. FCC, http://transition.fcc.gov/oet/tac/TACJune2011mtgfullpresentation.pdf, Meeting presentation of the Technological Advisory Council (2011b)

36. R. Lefebvre, P. Gournay, R. Salami, A study of design compromises for speech coders in packet networks, in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. I (2004) pp. 265–268
37. V. Eksler, M. Jelinek, Glottal-shape codebook to improve robust-ness of CELP codecs. *IEEE Trans. Audio Speech Lang. Process.* **18**(6), 1208–1217 (2010)
38. J.-M. Valin, K. Vos, T. Terriberry, Internet Engineering Task Force RFC6716 (2012)
39. S.V. Andersen, W.B. Kleijn, R. Hagen, J. Linden, M.N. Murthi, J. Skoglund, iLBC-A linear predictive coder with robustness to packet losses, in *IEEE Speech Coding Workshop Proceedings* (2002), pp. 23–25
40. T. Ogunfunmi, M.J. Narasimha, *Principles of Speech Coding* (CRC, BocaRaton, 2010)
41. K. Seto, T. Ogunfunmi, Multi-rate iLBC using the DCT, in *Proceedings of the IEEE Workshop on SiPS* (2010), pp. 478–482
42. K. Seto, T. Ogunfunmi, Performance enhanced multi-rate iLBC, in *Proceedings of the 45th Asilomar Conference* (2011)
43. K. Seto, T. Ogunfunmi, Scalable multi-rate iLBC, in *Proceedings of IEEE International Symposium on Circuits and Systems* (2012)
44. K. Seto, T. Ogunfunmi, Scalable speech coding for IP networks: beyond iLBC. *IEEE Trans. Audio Speech Lang. Process.* **21**(11), 2337–2345 (2013)
45. K. Seto, T. Ogunfunmi, Scalable wideband speech coding for IP networks, in *Proceedings of the 46th Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove (2012)
46. K. Seto, T. Ogunfunmi, A scalable wideband speech codec based on the iLBC, submitted to *IEEE Transactions on Audio, Speech, and Language Processing*
47. S.V. Andersen et al., Internet low bit-rate codec (iLBC) [Online]. RFC3951, IETF organization (2004), http://tools.ietf.org/html/rfc3951
48. C.M. Garrido, M.N. Murthi, S.V. Andersen, On variable rate frame independent predictive speech coding: re-engineering iLBC, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* **1**, 717–720 (2006)
49. J. Princen, A. Bradley, Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Trans. Acoust. Speech Signal Process.* **34**(5), 1153–1161 (1986)
50. ITU-T Rec. P.862, Perceptual Evaluation of Speech Quality (PESQ) (2001)
51. ITU-T Rec. P.501, Test signals for use in telephonometry (2012)
52. ITU-T Rec. G.191, Software tools for speech and audio coding standardization (2010)
53. E.N. Gilbert, Capacity of a burst-noise channel. *Bell Syst. Tech. J.* **39**, 1253–1265 (1960)
54. I. Daubechies, Orthonormal bases of compactly supported wavelets. *Commun. Pure Appl. Math.* **41**, 909–996 (1988)
55. F. Chen, K. Kuo, Complexity scalability design in the internet low bit rate codec (iLBC) for speech coding. *IEICE Trans. Inf. Syst.* **93**(5), 1238–1243 (2010)
56. D. Collins, *Carrier-Grade Voice-over-IP*, 2nd edn. (McGraw-Hill, New York, 2002)
57. A. Das, E. Paksoy, A. Gersho, Multimode and variable-rate coding of speech, in *Speech Coding and Synthesis*, ed. by W.B. Kleijn, K.K. Paliwal (Elsevier, Amsterdam, 1995), pp. 257–288
58. J. Davidson, *Voice-over-IP Fundamentals*, 2nd edn. (Cisco, Indianapolis, 2006)
59. G.D. Forney, Coset codes. I. Introduction and geometrical classification. *IEEE Trans. Inf. Theory* **34**(5), 1123–1151 (1988)
60. A. Gersho, Advances in speech and audio compression. *Proc. IEEE* **82**, 900–918 (1994)
61. J. Gibson, Speech coding methods, standards and applications. *IEEE Circuits Syst. Magazine* **5**(4), 30–40 (2005)
62. J. Gibson, J. Hu, *Rate distortion bounds for voice and video*, Foundations and Trends in Communications and Information Theory **10**(4), 379–514 (2013), http://dx.doi.org/10.1561/0100000061, ISBN: 978-1-60198-778-5
63. L. Hanzo, F.C.A. Somerville, J.P. Woodard, *Voice and Audio Compression for Wireless Communications*, 2nd edn. (Wiley, Chichester, 2007)
64. O. Hersent, *IP Telephony: Deploying VoIP Protocols and IMS Infrastructure* (Wiley, Chichester, 2010)

65. K. Homayounfar, Rate adaptive speech coding for universal multimedia access. *IEEE Signal Process. Magazine* **20**(2), 30–39 (2003)
66. ITU-T Rec. G.718, Frame error robust narrow-band and wideband embedded variable bit-rate coding of speech and audio from 8–32 kbit/s, International Telecommunication Union (ITU) (2008)
67. M. Jelinek et al., G.718: a new embedded speech and audio coding standard with high resilience to error-prone transmission channels. *IEEE Commun. Magazine* **46**(10), 117–123 (2009)
68. W.B. Kleijn, Enhancement of coded speech by constrained optimization, in *Proceedings of the IEEE Speech Coding Workshop* (2002)
69. J. Makinen, B. Bessette, S. Bruhn, P. Ojala, R. Salami, A. Taleb, AMR-WB+: a new audio coding standard for 3rd generation mobile audio services, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* **2**, 1109–1112 (2005)
70. S. Ragot, B. Bessette, R. Lefebvre, Low-complexity multi-rate lattice vector quantization with application to wideband speech coding at 32 kbit/s, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* **1**, 501–504 (2004)
71. M.R. Schroeder, B.S. Atal, Code-excited linear prediction (CELP): High-quality speech at very low bit rates, in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing* (1984), pp. 937–940
72. D. Wright, *Voice-over-Packet Networks* (Wiley, Chichester, 2001)

Chapter 4
Recent Speech Coding Technologies and Standards

Daniel J. Sinder, Imre Varga, Venkatesh Krishnan, Vivek Rajendran, and Stéphane Villette

Abstract This chapter presents an overview of recent developments in conversational speech coding technologies, important new algorithmic advances, and recent standardization activities in ITU-T, 3GPP, 3GPP2, MPEG and IETF that offer a significantly improved user experience during voice calls on existing and future communication systems. User experience is determined by speech quality, hence network operators are very concerned about quality of speech coders. Operators are also concerned about capacity, hence coding efficiency is another important measure. Advanced speech coding technologies provide the capability to both improve coding efficiency and user experience. One option to improve quality is to extend the audio bandwidth from traditional narrowband to wideband (16 kHz sampling) and super-wideband (32 kHz sampling). Another method is in increasing the robustness of the coder against transmission errors. Error concealment algorithms are used which substitute the missing parts of the audio signal as far as possible. In packet-switched applications (VoIP systems), special mechanisms are included in jitter buffer management (JBM) algorithms to maximize sound quality. It is of high importance to ensure standardization and deployment of speech coders that meet quality expectations. As an example of this, we refer to the Enhanced Voice Services (EVS) project in 3GPP that is developing the next generation speech coder in 3GPP. The basic motivation for 3GPP to start the EVS project was to extend the path of codec evolution by providing super-wideband experience at around 13 kb/s and better quality for music and mixed content in conversational applications. Optimized behavior in VoIP applications is achieved through the introduction of high error robustness, jitter buffer management, inclusion of source-controlled variable bit rate operation, support of various audio bandwidths, and stereo.

D.J. Sinder (✉) • V. Krishnan • V. Rajendran • S. Villette
Qualcomm Technologies, Inc., 5775 Morehouse Dr., San Diego, CA 92121, USA
e-mail: dsinder@qti.qualcomm.com; vkrishna@qti.qualcomm.com; vivekr@qti.qualcomm.com; svillett@qti.qualcomm.com

I. Varga
QUALCOMM CDMA Technologies GmbH, Franziskaner Str. 14, D-81669, Munich, Germany
e-mail: ivarga@qti.qualcomm.com

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_4

4.1 Recent Speech Codec Technologies and Features

This section presents key functional blocks in speech coders which apply advanced technology from today's perspective. Among them, we can find techniques to exploit the nature of speech signals allowing different coding methods for active speech and pauses combined with discontinuous transmission, or even allowing source-controlled variable bit rate operation. In achieving enhanced quality and user experience through extended audio bandwidth over traditional narrowband, bandwidth extension algorithms play a key role. They generate missing higher bandwidth signal (e.g., wideband) from available narrowband data (blind bandwidth extension) or exploit the structural dependencies between low and high-bands and transmit additional information for the high-band as part of the bit stream. The concept of layered coding is presented as the key for scalable coding. Also packet loss concealment algorithms are described which are especially important in error prone environment, typically mobile networks, VoIP and Internet.

4.1.1 Active Speech Source-Controlled Variable Bit Rate, Constant Bit Rate Operation and Voice Activity Detectors

Voice activity detection (VAD) is a technique employed in speech coders wherein the presence of speech is detected. The identified speech regions—termed, *active speech*—are thereby separated from background noise, or *inactive*, segments. This enables discontinuous transmission (also known as DTX), in which transmission is temporarily cut off in the absence of active speech. During DTX, the speech encoder and transmitter cease continuous encoding and transmission, but transmits "silence indicator" (SID) frames resulting in lower average power consumption in mobile handsets. SID frames are coded at a significantly lower bit rate than active speech which also enables lower average data rate and higher capacity. These SID frames are used by the speech decoder/receiver's comfort noise generation (CNG) modules to synthesize background noise at the decoded output.

A typical VAD works as follows: the input speech frame (typically 20 ms) samples are split into frequency sub-bands or critical bands. The signal level and an estimate of the background noise level in each frequency band are computed. The background noise level estimate depends on previous VAD decisions and signal characteristics such as stationarity and tonality. The input signal-to-noise ratio is compared to an adaptive threshold to compute an intermediate VAD decision. Threshold adaptation can be done depending on the desired sensitivity and is based on noise and long term speech estimates. The final VAD decision is calculated by adding hangover to the intermediate VAD decision. The hangover addition helps to detect the trailing low energy ending of words which are subjectively important but difficult to detect.

A robust VAD (especially on a mobile phone) must be able to reliably detect speech in the presence of a range of very diverse types of ambient noise and levels to achieve the balance between capacity and quality. Low signal-to-noise ratio conditions present particularly difficult detection conditions and increase the probability of false positives i.e. speech detected as background noise which may result in speech clipping.

4.1.1.1 Source-Controlled Variable Bit Rate (SC-VBR) Versus Constant/Fixed Bit Rate (CBR) Vocoders

SC-VBR vocoders select the speech encoding rate and mode based upon the characteristics of the input speech—e.g., voiced, unvoiced, transient, and stationary. In particular stationary voiced and unvoiced (fricative) speech can be encoded at lower bit rates with marginal impact to voice quality as compared to transient or non-stationary speech. This introduces the capability to operate at multiple capacity operating points (COPs) each with different active speech average data rates. These COPs can be used to target varying system capacity while trading off speech quality gracefully. The encoder capacity operating point can be controlled by the network based on network congestion or other factors. For example, the EVRC-B vocoder (a narrowband speech codec used for CDMA2000 systems described in Sect. 4.2.4) attempts to meet the specified target active speech average data rate by adjusting the proportion of low rate and high rate active speech frames. This adjustment is done dynamically by comparing the actual average data rate in a window of past active speech frames to the target active speech average data rate, and computing an appropriate fraction of low rate and high rate frames for a window of future active speech frames.

It is important to distinguish the active speech source controlled variable bit rate operation from active/inactive speech variable bit rate operation employing VAD/DTX/CNG mechanisms. In contrast to SC-VBR vocoders, constant bit rate vocoders like AMR (the 3rd Generation Partnership Project (3GPP) narrowband speech codec used for GSM/UMTS systems) operate at a fixed rate for every active speech frame. However they can support multiple such fixed data rates which are network controlled for dynamic adaptation to network conditions. For example, the network can switch to using lower bit rates during network congestion to improve capacity or perhaps to trade off speech bit rate for channel coding to increase channel protection.

Table 4.1 classifies some of the vocoder standards by active speech variable versus constant (fixed) bit rate. More details on these Standards Development Organizations (SDOs) and codecs are presented in Sect. 4.2.

Table 4.1 Standardized conversational coders and their respective standards developing organization (SDO), separated by whether they are constant bit rate or active speech variable bit rate

	SDO	Coders
Constant bit rate	ETSI	GSM-FR, GSM-HR, GSM-EFR
	3GPP	AMR, AMR-WB, EVS (some modes)
	ITU-T	G.711, G.711.0, G.711.1, G.722, G.722.1, G.722.1C, G.722.2, G.718, G.719, G.729, G.729.1
	IETF	iLBC
	ISO/IEC MPEG	AAC-LD, AAC-ELD, AAC-ELD v2
Active speech variable bit rate	3GPP 3GPP2	EVS (some modes) EVRC, EVRC-B, EVRC-WB, EVRC-NW, EVRC-NW2K, VMR-WB
	IETF	Opus
	N/A	Speex, SILK

4.1.2 Layered Coding

Today, communication technology migrates toward VoIP networks and enhanced user experience through wideband, super-wideband, and full-band speech which extend audio bandwidth over traditional narrowband speech. Earlier narrowband and wideband coders are incompatible in the sense that a wideband coder does not extend a narrowband coder. In other words, a decision on audio bandwidth is needed before encoding. Layered coding allows the feature of bandwidth scalability which is very useful in the sense that it makes switching the bandwidth possible. A further aspect is interoperability due to the interconnection of various communication networks using different speech coding standards. Interoperability issues can be effectively minimized by layered extension of existing coders rather than introducing new incompatible coders.

Scalability can be used for several purposes. By building upon a core layer that is bit stream interoperable with a standard narrowband coder, scalability can be used to improve the legacy coder while still maintaining interoperability through removing or ignoring the enhancement layers in the bit stream. Alternatively or in combination with legacy coder enhancement, layers can be added to the core to improve coding resolution (i.e., reduce quantization noise), extend the audio bandwidth, or improve robustness against transmission errors.

Layered coding is a concept in which a core layer is overlaid with multiple enhancement layers. The core layer provides the minimum number of bits needed by the decoder for re-synthesizing the speech signal with a minimum (core) quality. Additional bits to improve quality are then added in layers such that each layer, starting with the core layer, is embedded in the overall bit stream. Figure 4.1 illustrates the concept of layered coding where a core layer is extended by multiple enhancement layers.

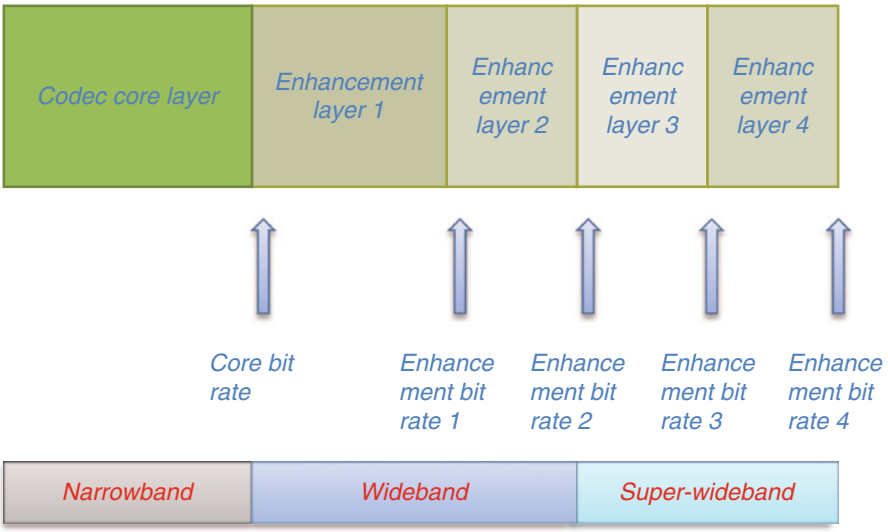

Fig. 4.1 Illustration of layered coding

A typical situation is that the core layer represents narrowband coding at a certain bit rate while the enhancement layers add gradual improvements at the expense of additional bit rate. Improvements provide enhanced user experience due to improved coding within the same audio bandwidth or due to extended audio bandwidth. Speech coders implementing this type of layered structure have been standardized in ITU-T and they will be described in the standardization part of this chapter.

4.1.3 Bandwidth Extension of Speech

Traditional wireline and wireless telecommunication systems carried voice that was limited to 4 kHz in bandwidth. Termed "narrowband voice", this band limited speech signal had quality and intelligibility good enough to sustain a two way conversation, but lacked the richness and the fullness of natural human voice. The bandwidth limitation in traditional telephony networks was largely due to the presence of PSTN sub-systems that are inherently narrowband. As wireless voice communication systems continued to evolve, networks moved toward transcoder-free operation where no PSTN subsystem was involved. Further, the recent emergence of voice over mobile broadband has enabled end to end voice transportation without the need for intermediate transcoding. These, coupled with advances in the electro-acoustic capabilities on wireless devices, have broadened the scope for employing speech coding techniques that encode speech signals with bandwidths wider than narrowband voice. Wideband vocoders can encode signal

bandwidths ranging from 50 Hz to nearly 8 kHz, while super-wideband vocoders extend the upper range of the coded bandwidth up to 16 kHz.

Coding voice bandwidths wider than narrowband at bit rates that are comparable to narrowband coders has largely been made possible by bandwidth extension technologies. The inaccuracies in the representation of the spectral and the temporal information content at higher frequencies in a speech signal are masked more easily than contents at lower frequencies. Consequently, bandwidth extension methodologies manage to encode the spectral regions beyond the narrowband frequency range in speech signals with far fewer bits than what is used to code signal content in the narrowband frequency range. In coding the higher frequency bands that extend beyond the narrowband frequency range, bandwidth extension techniques exploit the inherent relationship between the signal structures in these bands. Since the fine signal structure in the higher bands are closely related to that in the lower band, explicit coding of the fine structure of the high band is avoided. Instead, the fine structure is extrapolated from the low band. Then the correction factors that are needed to modify the extrapolated fine structure are then transmitted from the encoder to enable the decoder to reconstruct speech whose bandwidth is wider than that of narrowband speech. The correction factors to be transmitted are chosen to trade off the quantity of bits needed to encode these correction factors with the need to perceptually mask the inaccuracies in representing the high band signal content of the signal.

The sections below present an overview of bandwidth extension techniques that are widely used in various vocoders used for wireless and wireline communications today.

4.1.3.1 Harmonic Bandwidth Extension Architecture

In bandwidth extension methodologies that seek to extend the bandwidths of signals coded by a core narrowband vocoder that is based on an LPC paradigm, the high band excitation signal is derived from the low band excitation in the form of the excitation signal provided by the narrowband coder. To preserve the harmonic structure of the low frequency excitation signal in the high frequency excitation signal, a non-linear function (absolute value) is used [39]. The nonlinear function is applied after sufficiently over-sampling the narrowband signal in order to minimize aliasing. Fixed or adaptive whitening can be applied to the output of the nonlinear function to flatten the spectrum. The main drawback of the nonlinear function is that for many voiced speech signals, the lower frequencies exhibit a stronger harmonic structure than higher frequencies. As a result, the output of the nonlinear function can lead to a high frequency excitation signal that is too harmonic, leading to objectionable, 'buzzy'-sounding artifacts [36]. As a solution, a combination of a nonlinear function and noise modulation is used to produce a pleasantly-sounding high-band signal.

Figure 4.2 depicts the process that generates the high band excitation from the narrowband excitation signal. The excitation signal is first run through an

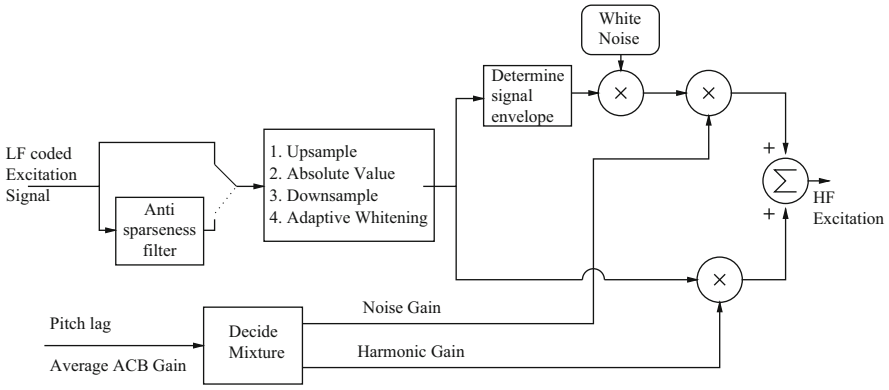

Fig. 4.2 Generation of high band excitation signal

all-pass filter. This filter reduces the sparseness that results from encoding the low-band signal with a sparse fixed codebook, and is intended to be used during unvoiced speech. Next is the nonlinear function. This module up-samples the signal to 64 kHz, takes the absolute value, and then down-samples it to 16 kHz. From here, a 7 kHz-sampled signal is produced using the same high-band analysis filter that was used to split the input signal in a low and a high band. The result is spectrally flattened with an adaptive 4th order linear prediction filter, to create the harmonically-extended excitation signal.

A modulated noise signal is generated by multiplying a unit-variance white noise signal with the envelope of the harmonically-extended excitation signal. This envelope is obtained by taking the squared value of each sample, smoothing with a first order IIR low-pass filter and taking the square-root of each smoothed sample. The modulated noise and harmonically-extended excitation signals are now mixed together to create a signal with the right amount of harmonic and noise contents.

4.1.3.2 Spectral Band Replication (SBR)

Spectral Band Replication (SBR) [38] is a bandwidth extension technology that is often used with perceptual audio codecs such as MP3 and AAC to enable low bit rate coding of audio signals. SBR codes the high frequency components of an audio signal by transposing the low frequency components of the signal to the high frequency region and using a set of adjustment parameters that indicate the modifications to the transposed low frequency components to match up the high frequency components of the input signal.

The SBR encoder works in conjunction with a core encoder that encodes the low frequency portions of an input signal. The input signal is first filtered through a bank of bandpass filters and the core encoder is invoked to code the low frequency sub-band(s). Typically Quadrature Mirror Filters (QMF) are employed for decomposing

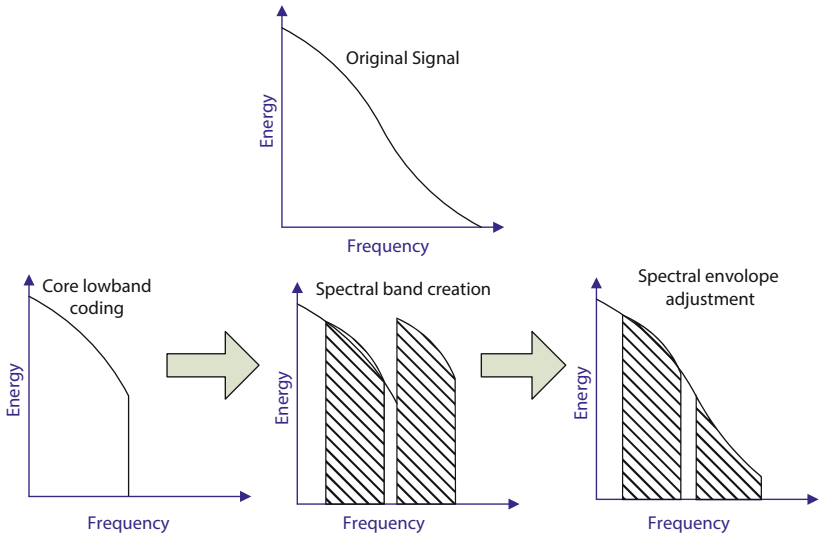

Fig. 4.3 Generation of high band excitation signal using SBR

the input signal into sub-bands. At the decoder, the SBR decoder transposes the low frequency components obtained from the core decoder to the higher frequency range through a simple translation to high frequencies as shown in Fig. 4.3. The SBR encoder transmits several adjustment parameters which are then used by the SBR decoder to modify the transposed high frequency components to reproduce the high frequency components of the input signal. The adjustment parameter extraction algorithm in an SBR encoder is tuned to the core coder at a given bit rate and sampling rate. The adjustment parameters include spectral envelopes of the high band components of the input signal. The temporal and spectral resolution of these envelopes are chosen according to the characteristics of the input signal and are adapted every frame. Special considerations are included for transient signals where the spectral content in the original signal is largely concentrated in the high band. For such transients, the time-frequency resolution of the envelope is chosen to represent the non stationary nature of this signal within a given frame. Besides transients, special cases such as harmonic low bands and noise-like high bands are handled in SBR.

4.1.4 Blind Bandwidth Extension

Blind Bandwidth Extension (BBE) is the generic term used to describe technologies that are able to predict from a given band-limited audio signal (e.g., a narrowband signal with frequency content between 0 and 4 kHz) into an audio signal of higher

bandwidth (e.g. a wideband signal with content between 0 and 8 kHz), without requiring the transmission of additional data (hence the term "blind"). The challenge for these technologies is to generate new frequency components that approximate the original input such that the resulting signal retains high quality and intelligibility without introducing annoying artifacts.

4.1.4.1 High Band Model and Prediction Methods

Typically, a BBE system consists of two modules. One module is a bandwidth extension module, or high band model, which generates the missing frequencies in the high band based upon the input signal and some additional input parameters. The other module is a prediction module which attempts to predict those additional parameters from the input signal. The high band model may, for example, be similar to that described in Sect. 4.1.3.

The prediction module generally extracts some features from the input signal (e.g. spectral shape, pitch, gain), and attempts to predict parameter inputs to the high band model. This can be done using heuristics, or more commonly, statistical modeling of the joint distribution of the input features and high band parameters. This statistical modeling can be performed for example using VQ, GMM, HMM, or other appropriate means.

4.1.4.2 BBE for Speech Coding

One popular application of BBE is within the context of a speech coder. In certain cases, it may be more efficient for a speech coder to discard parts of the input signal completely, and then attempt to blindly regenerate them at the decoder, rather than encoding these parts in a traditional manner. This is particularly true if the frequency band that is discarded is small, and contains relatively little extra information over the transmitted content.

AMR-WB [11] uses such a technique: the 16 kHz-sampled input signal contains frequencies up to 8 kHz, but only the 0–6.4 kHz band is coded. The frequencies above 6.4 kHz are predicted using a simple model, where a random noise source is spectrally shaped using a filter extrapolated from the LPC filter used to code the 0–6.4 kHz band, and a gain is predicted using the voicing strength of the 0–6.4 kHz band. This produces acceptable quality, and allows the coding effort of the ACELP core to be focused on the perceptually more important part of the spectrum, below 6.4 kHz.

AMR-WB+ [40] also uses a similar technique, although in this case a small number of bits (0.8 kb/s) are transmitted to correct for gain errors made by the BWE module. The 23.85 kb/s mode of AMR-WB similarly uses 0.8 kb/s for gain correction.

4.1.4.3 BBE for Bandwidth Increase

In certain circumstances, it can be desirable to increase the bandwidth of a speech signal. This might be done either because the increased bandwidth of the resulting signal might be perceptually more pleasant, or to attempt to minimized distortions that would occur if the bandwidth of the signal was to vary.

While the world of telephony used exclusively narrowband signals until fairly recently, there are now practical solutions for wideband and super-wideband speech communications. Most prominently, the AMR-WB and EVRC-NW wideband codecs have been commercially deployed on numerous networks, and the EVS codec currently under standardization by 3GPP will provide super-wideband modes. However, these are not yet ubiquitously available, and a user is likely to experience varying levels of bandwidths depending on the network he and his interlocutor are using, as well as the equipment used by the other side. Bandwidth switching may even occur during a given call, which is very detrimental [46].

Currently there are a number of bandwidth extension solutions available commercially, but deployment remains limited at this time, and no standards body has standardized a BBE solution to date. However, with the emergence of speech coders able to transmit ever increasing speech bandwidths and the detrimental impact of bandwidth switching on user experience, it is reasonable to expect that BBE solutions will become more wide-spread, and that deployment of BBE will go hand-in-hand with that of higher bandwidth codecs.

4.1.4.4 Quality Evaluation

Speech processing or coding systems are usually evaluated by comparing the original input, which is considered to be the quality reference, with the output. Typically the output is degraded by the processing, and the difference can be evaluated using either objective (e.g. PESQ, POLQA), or subjective (e.g. ITU P.800 ACR/DCR, MUSHRA) testing.

Bandwidth extension technology is fundamentally different in that it does not try to maintain the quality of the input signal, but tries to improve it, and most importantly, change its nature. As a result, there is currently no objective measure that can adequately assess the quality of BBE. The objective measures to evaluate a wideband speech quality that are most prominently used in the telecommunications industry today require a wideband reference. But in the case of BBE, there is no wideband reference, only a narrowband reference. It is of course tempting to start from a wideband signal, down-sample it to narrowband, run the BBE on the narrowband signal, and compare its output with the original wideband signal. However this is fundamentally flawed: BBE attempts to generate a wideband signal with good quality, but it has no way of knowing what the original high band was like. Therefore, it may well generate a high quality wideband signal, that differs significantly from the original. Objective measures would score it down because of the mismatch, even though the quality may be good, and possibly better than another

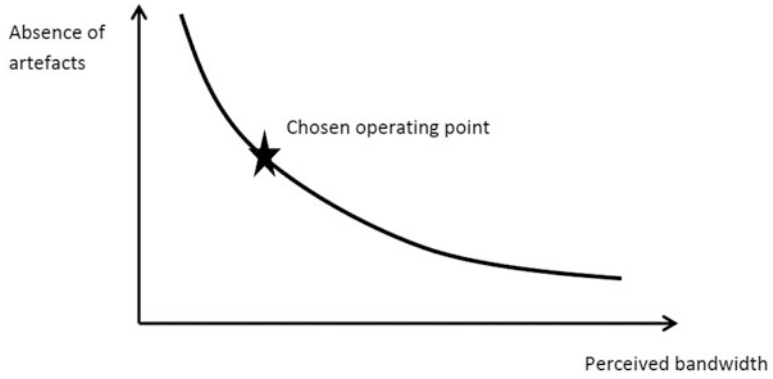

Fig. 4.4 Trade-off between bandwidth and artifacts

BBE algorithm that produces low quality speech but a better match to the original. (Note that objective measures, such as PESQ and POLQA, should not be used for this purpose, as their specifications explicitly state that they are not necessarily valid outside their design and training envelope, which does not cover BBE.)

One more difficulty is the terminal frequency response specification. It is tempting to expect BBE to fit the same Rx frequency response as a wideband codec. However, there are two problems with this. The first is a fundamental problem that the transfer function is not defined since the input has a smaller bandwidth than the output. The second problem is that BBE typically introduces artifacts, and lowering the energy in the high band can lead to a better compromise between extended bandwidth perception and speech quality. When comparing BBE algorithms, it is important to understand this trade-off, and only compare algorithms at the same tuning point—e.g., either similar artifacts or similar bandwidths. This is illustrated in Figs. 4.4 and 4.5.

Since one of the main aims of BBE is to reduce the impact of bandwidth switching, the ITU P.800 DCR is a reasonable way to evaluate BBE algorithm quality. It is unfortunately a rather complicated and expensive test to run, but in the absence of suitable objective measures, it is currently the best available test for BBE evaluation to our knowledge.

4.1.4.5 Encoder Based BBE

Aside from deploying a full new WB codec, or using a BBE technique on top of an existing NB codec, an approach that has been proposed previously consists of developing a WB codec which is fully backward compatible with the existing NB infrastructure [13]. The idea is to take a WB speech signal, and split it into a NB signal (typically 0–4 kHz), and a high band (HB) signal (4–8 kHz). The HB signal can be coded efficiently using a small number of bits, as it typically contains

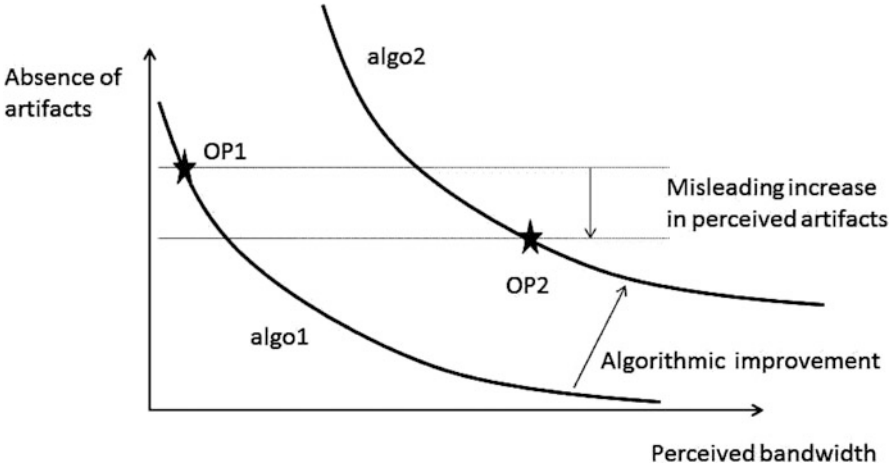

Fig. 4.5 Misleading performance comparison

much less information than the NB signal, and is highly correlated with the NB. The NB is coded with a standard NB codec, while the HB information is hidden in the NB bit stream using watermarking techniques. Watermarking, also known as steganography, consists of hiding a data stream within another data stream. The concept itself is very ancient, and has been applied to various media forms, from text to images, videos, and audio. Here, we are interested in hiding data within a compressed bit stream. This can be done by constraining the bit patterns that can be transmitted by a codec, in a way that can be detected and decoded at the receiving end. Simple techniques exist, such as splitting a quantization table into two half-tables, and using indexes from the half-table corresponding to the bit of hidden data that is being sent. Provided the tables are well split, typically so that they both adequately cover the codeword space, information can be hidden with relatively little quality loss [13].

This approach can be thought of as Encoder-based BBE (or Tx-side BBE), as the BBE is performed at the decoder, but using parameters that are transmitted by the encoder. The advantage over the more common Decoder-based BBE (or Rx-side BBE) is that the encoder has the original WB input available, hence there is no risk of wrong parameter estimation. This leads to significantly improved quality, comparable to existing dedicated WB coders such as AMR-WB or EVRC-NW.

To provide wideband speech, the decoder must receive the same bits that were sent by the encoder. This may not always be the case in a mobile telephony system. Indeed, it is the norm that the transmitted packets will be decoded to PCM in the network, and then re-encoded before being sent to the decoder. As this is inefficient, and degrades quality, Transcoder Free Operation (TrFO) has been deployed on some UMTS networks, ensuring that the coded speech bit stream does not get re-encoded over the network. TrFO is currently deployed on some networks, and is becoming increasingly more common as it is more efficient in terms of capacity.

The equivalent for GSM networks, called TFO, is also available in some networks. Overall, Tx-side BBE provides the same user experience as the current NB codec in the cases of mobile to wireline calls, or mobile to mobile when TrFO is not present. However, for mobile-to-mobile calls when TrFO is present, WB speech will be delivered. Note that it is very possible, and indeed probably best, to combine Tx-side BBE with Rx-side BBE. They may even share the same high band model, and only vary in the way the high band parameters are determined. In this case, the decoded speech signal will always be WB, but with increased quality when both ends support Tx BBE.

A watermarking codec has some disadvantages over a conventional WB codec. The fact that it is backward compatible with a pre-existing NB codec puts a lots of constraints on its design, and therefore it tends to have a slightly lower coding efficiency than a WB codec free of that constraint, at equal level of technology. However, in cases where only the watermarking codec can operate (because the conventional WB codec has not been deployed), then this comparison is irrelevant, and the comparison should be with the existing NB codec. Another risk is that a badly designed watermarking scheme may introduce too much noise, and cause degradation in the legacy case. This is not acceptable, and the watermarking scheme must be such that its impact on NB quality is negligible. The watermarking approach however has significant advantages. For example, by not requiring any network changes, the only cost is that of deploying new software in the handsets, and ensuring the electro-acoustics work well with WB. Additionally, by not requiring a new codec to be deployed, it automatically ensures that there will be no problems interoperating with the other existing codecs and terminals. This has proved to be a concern when deploying AMR-WB.

Due to the full legacy interoperability of encoder-based BBE schemes, it is difficult to know what may have been deployed in terminals as proprietary technology. However, there is at least one known commercially available encoder-based BBE solution offered by Qualcomm, and marketed under the name of eAMR (for enhanced-AMR). It has been demonstrated to work well at major industry trade shows over live commercial networks, confirming that encoder-based BBE can be a practical solution to help increase the footprint of WB in mobile telephony.

4.1.5 Packet Loss Concealment

Frame erasures or packet loss can have a significant impact to voice quality. Hence it is critical for voice codecs deployed over wireless communication systems or voice over packet switched networks to have efficient frame erasure concealment mechanisms. We address packet loss concealment algorithms used for CELP and ADPCM based coding next.

4.1.5.1 Code Excited Linear Prediction Coders

Code Excited Linear Prediction (CELP) is a technique used on most recent low bit rate speech coding standards. While the pitch predictive component or the adaptive codebook significantly contributes toward achieving high speech quality at low bit rates, it also introduces sensitivity to frame loss due to dependency on information from past frames. When a frame of speech is lost due to erasure, the adaptive codebook parameters and other relevant parameters are extrapolated from the previous frame to synthesize an output at the decoder. But the content of the adaptive codebook is different from that at the encoder which introduces mismatch or de-synchronization. This causes the synthesized subsequent good frames which are received after the lost frame, to deviate from the case if there was no loss.

4.1.5.1.1 Fast Recovery

Fast recovery is an approach where side information with some bit rate overhead is transmitted to arrest error propagation into future frames, thereby improving performance under frame erasures. Side information can include parameters like energy, frame classification information and phase information. The phase information can be used to align the glottal pulse position at the decoder to that of the encoder thereby synchronizing the adaptive codebook content. If side information (such as the phase information) is not transmitted, the resynchronization is done based on a predicted pitch value of the lost frame. The prediction is based on past pitch values and/or the future frame pitch (if available). These techniques are employed in VMR-WB (3GPP2), G.729.1 (ITU-T) and G.718 (ITU-T) speech coding standards.

There can also be instances where the future frame containing the side information is not available at the time of synthesizing the lost frame to avoid adding extra delay at the decoder. The information on the lost frame which becomes available on receiving the future frame can be used to correct the excitation (pitch) memory before synthesizing the correctly received future frame. This helps to significantly contain the error propagation into future frames and improves decoder convergence when good frames are received after the erased frame. Waveform interpolation techniques are necessary to avoid abrupt changes in the pitch contour between the error concealed lost frame and the memory corrected future frame.

4.1.5.1.2 Loss of Voiced Onsets

Voiced onset frames are typically preceded by inactive or unvoiced speech frames, which lack the periodic component in the excitation signal. The frame following the voiced onset relies heavily on the periodic component in the voice onset frame to encode the waveform. Since the periodic component of the excitation is completely missing when there is a loss of the voiced onset frame, it can take several frames to recover and potentially suppress the entire vowel sound.

Artificial onset construction is a technique used in 3GPP2 VMR-WB codec where the adaptive codebook of the first correctly received frame following the voiced onset is synthesized by low pass filtering an impulse or two pulses separated by an appropriate pitch period followed by regular decoding. Side information from a future frame containing the position and sign of the last glottal pulse in the erased frame can also be used for the artificial reconstruction.

Transition coding mode is a technique used in the ITU-T G.718 codec which alleviates the impact of voiced onset loss by selecting a code vector from a glottal shaped codebook to encode the adaptive codebook component. This removes the dependency on the excitation signal from the previous frame thereby eliminating the primary cause of frame error propagation.

4.1.5.2 Adaptive Differential Pulse Code Modulation (ADPCM) Based Coders

ADPCM coding is highly recursive. For example, the G.722 coder uses embedded ADPCM with 6, 5 or 4 bits per sample to code the low band (0–4 kHz) and 2 bits per sample to code the high band. The quantization scale factor, Moving Average (MA) and Auto-regressive (AR) prediction coefficients are updated on a per sample basis. Loss of synchronization between the encoder and decoder needs to be handled carefully to avoid artifacts due to frame erasures.

Packet loss concealment technique for a typical sub-band ADPCM coder like G.722 is described as follows. Linear Predictive Coding (LPC) analysis is performed on the past frame low band synthesis signal. The resulting LP residual signal is used for Long Term Prediction (LTP) analysis to estimate an open loop pitch delay. Pitch synchronous period repetition of the past LP residual signal is performed. Signal classification information is used to control the pitch repetition procedure. The low band extrapolated signal is obtained by filtering the resulting excitation signal through the LPC synthesis filter. The extrapolated low band signal is used to update the ADPCM decoder state memories. Cross fading is also performed to ensure a smooth transition between the extrapolated samples in the lost frame and the initial few samples of the first good frame. High band concealment simply consists of pitch synchronous repetition of the past high band output signal controlled by the signal classification information.

4.1.6 Voice Over Internet Protocol (VoIP)

The ubiquity of high speed packet switched networks, and the specifically the Internet, brought with it the possibility of voice communications between any two, or more, Internet connected devices with suitable audio sound capture and rendering equipment. Today's VoIP system are built upon a collection of protocols and standards enabling successful private and public networks and services. These standards

include transport protocols such as the Real-time Transport Protocol (RTP) [41], as well as signaling and control protocols such as Session Initiation Protocol (SIP) [14] or H.323 [35].

Since this chapter's focus is on recent speech coding technologies and standards, the focus in this section is not on these, and many other protocols and standards upon which VoIP systems are built. Instead, this section will describe recent technologies for VoIP that are typically integrated into the speech coder itself or possibly realized through interaction between the coder and the VoIP client, which is the VoIP call session manager.

VoIP technologies for coders are designed to accommodate idiosyncrasies of packet switched networks that were rarely, if ever, evident in circuit switched networks, and that force conventional coder designed for circuit switched networks to operate outside of their design envelope. The two most impactful characteristics creating problems for conventional coders are time-varying delay (or, delay jitter) and bursts of consecutive packet loss. Furthermore, due to the use of VoIP on general purpose data networks without quality of service (QoS) management, rates of packet loss—either due to excessive delay or transport loss—can be considerably higher than typically seen on circuit switched networks. Technologies that compensate for these characteristics help substantially to maintain voice quality for end users.

4.1.6.1 Management of Time Varying Delay

In packet switched networks, packets may be subjected to varying scheduling and routing conditions, resulting in time varying transit time from end to end. This time varying delay, or delay jitter, is not suitable for most conventional speech decoders and voice post-processing algorithms, which have historically been developed with the expectation that packets are transmitted and received on a fixed time interval. As a result VoIP clients utilize a buffer in the receiving terminal to remove jitter.

For conversational voice, mouth-to-ear delay is a key determiner of conversation call quality. If this delay gets too high, the interactivity of the talkers is impaired, and users can experience double-talk and overall dissatisfaction. The tolerance for high delay can be impacted by several factors, such as individual tolerance and the level of interactivity. ITU-T G.114 [16] provides what is probably the most widely accepted guideline for the relationship between mouth-to-ear delay and caller satisfaction. The receiver's ability to remove delay jitter without adding excessive delay for buffering is thus a key component to the management of mouth-to-ear delay and, hence, call quality.

At the same time, the longer the buffer for jitter removal, or de-jitter buffer, the greater the likelihood that high jitter conditions can be tolerated without dropping a high percentage of packets due to their arriving too late to be played out. Thus, a clear trade-off exists between the de-jitter buffering delay and the jitter induced packet loss at the receiver. A de-jitter buffer can employ several layers of delay management to achieve the desired operating point.

First, and simplest, is a fixed length de-jitter buffer. Such a tactic is really only suitable if the network's jitter properties are well characterized and known to be stable over time and varying network loads. Also, a fixed buffer length is only suitable if a length can be used that provides both satisfactory end-to-end delay and satisfactory voice quality, with consideration of jitter induced packet losses under that buffer length.

For most wireless packet data systems carrying VoIP traffic, a fixed length buffer is not sufficient, and another layer of sophistication is needed. Typically the next step is to adapt the buffer's length in between talk spurts (i.e., during silence). This offers the flexibility to grow the buffering delay only during periods of excessive jitter, but otherwise keep buffering delay low thereby reducing overall average delay. Also, by restricting buffer adaptation to silence periods, perceptible quality distortions are kept to a minimum.

For the same frame loss rate, even lower average delay can be achieved by dynamically adapting the play-out length of active speech frames. Such adaptation is known as time-scale modification, or time-warping [37, 47]. By making small time-scale adjustments to active speech, the receiver can more closely follow the instantaneous delay jitter changes. Yavuz et al., for example, describe a technique for time-warping that uses a target frame loss rate to control the adaptation [47]. Liang et al. show in [37] how the three levels of sophistication described here lead to increasingly closer tracking of instantaneous delay by the receiver, thereby minimizing the overall average delay.

4.1.6.2 Packet Loss Concealment for VoIP

The packet loss concealment techniques described in Sect. 4.1.5 are, of course, still relevant and applicable when those same codecs are used on packet-switched networks. However, the patterns of packet delays and losses, or delay-loss profiles, are generally different between circuit-switched and packet-switched networks. Even on packet-switched networks employing QoS mechanisms, multiple consecutive packet losses tend to be more common than the comparatively uniformly distributed losses seen on circuit switched networks. As a result, additional packet loss concealment technologies are desirable. Further, the de-jitter buffer at the receive offers an opportunity to employ not only new packet-loss concealment methods, but also packet-loss protection to prevent the loss at the decoder in the first place.

When a VoIP decoder needs to receive the next packet in order to produce a continuous sequence of output speech samples, the next packet must be available in the de-jitter buffer. If not, a buffer underflow occurs. Conventionally in these circumstances, the decoder enacts the packet loss concealment mechanisms noted in Sect. 4.1.5. However, since the de-jitter buffer, which comes with an associated cost in terms of delay, is a necessary evil for VoIP, it's presence should be leveraged for improved underflow prevention and more sophisticated packet loss concealment. This can be achieved using two techniques— interpolation and redundancy.

4.1.6.2.1 Interpolation

Interpolation based packet loss concealment (IPLC) replaces the more conventional extrapolation based loss concealment. During circuit-switched operation, since the packet sent subsequent to a lost packet is not available at the time of the loss, the lost information must be extrapolated from the past—usually from the previous frame. However, in the case of a late packet arrival in a packet-switched network, it is possible that, due to delay jitter, the subsequent packet has already arrived and is available in the de-jitter buffer. Therefore, a more smooth reconstruction of the lost packet can be achieved by interpolating between the previous packet and the following packet, rather than extrapolating from the past alone.

The benefit of IPLC over extrapolation techniques tends to increase with the number of consecutive lost packets, owing in part to error propagation that is characteristic of predictive coders (see Sect. 4.1.5). The longer the period of loss, the greater the deviation between the predictive parameters extrapolated from the past at the decoder and the true parameters computed at the encoder when encoding the first frame after the loss that is received at the decoder. The inherent smoothing of IPLC can thus reduce the occurrence of pops and clicks when decoding the first received frame after the loss.

The complication of developing IPLC algorithms comes from the fact that it is non-trivial to determine when to choose IPLC versus extrapolation. This is illustrated in Fig. 4.6. As shown at the to portion of the figure, delay jitter may cause a few consecutive packets to be delayed together, such that when decoding frame n, IPLC is not a clearly preferred choice because the next available packet may be several frames away. If conventional extrapolation based concealment is used in this case for frame n, more of the following frames may have arrived when the time to decode frame $n + 1$ arrives, and IPLC still can be used to smooth over the loss.

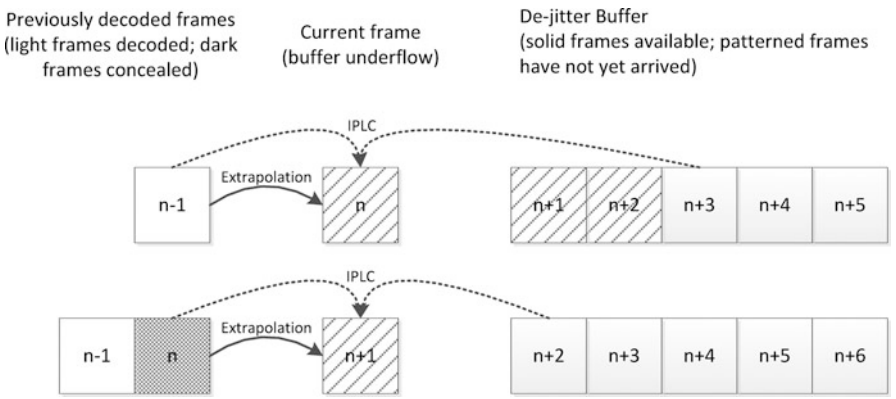

Fig. 4.6 An illustration of extrapolation and interpolation options for VoIP packet loss concealment. In the *top* example, extrapolation may still be preferred over interpolation due to the separation of packets available in the de-jitter buffer, while the reverse is true for the example on the *bottom* where only one missing packet separates the available packets

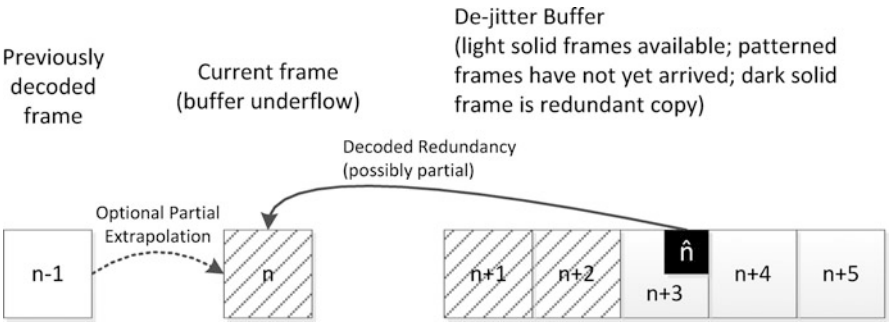

Fig. 4.7 An illustration of using a redundant copy of frame n in the de-jitter buffer for packet loss concealment

IPLC, of course, offers no benefit if the subsequent packet is not yet available in the de-jitter buffer. This is increasingly likely if the sequence of losses is long, and it is a certainty if the sequence of losses is longer than the buffer. Therefore, another mechanism is needed to reduce the loss burst length. This is one of the main benefits of using redundancy, described below.

4.1.6.2.2 Redundancy

For particularly long bursts of buffer underflow, even a small amount of information about the lost packets can be extremely helpful to conceal the loss. In extreme cases, a goal may be simply to preserve intelligibility of the decoded speech rather than completely concealing the loss. To this end, information about a frame can be transmitted redundantly along with a future frame. This is illustrated in Fig. 4.7.

As discussed above in connection with Fig. 4.6, IPLC over long underflow bursts is not always optimal. However, if frame $n + 3$ in the top portion of that same figure also carried information about frame n, as illustrated in Fig. 4.7, then frame n could be well concealed with parameters from the encoder, making the concealment of frame $n + 1$ in the lower portion Fig. 4.6 even higher quality.

There are numerous options for designing redundancy schemes. The redundant copy can be either a full copy of the original, or it can be a partial copy that includes just a subset of parameters that are most critical for decoding or arresting error propagation. In the case of a full copy, it may be desirable to encode it at a reduced rate, either with the same coding model or an entirely different model better suited for lower rate encoding.

The mechanism for transmitting redundancy can also vary. At the simplest level, redundancy can be included at the transport layer (e.g., by including multiple packets in a single RTP payload). This possibility, for example, is included in the design of the RTP payload format for AMR and AMR-WB [42]. This has the advantage that redundancy can be utilized with codecs that were originally designed for circuit switched operation, such as AMR and AMR-WB.

Alternatively, and more optimally, redundancy can be designed into new codecs. Such is the case for EVS, which is being designed with modes that include the option of transmitting redundancy in-band as part of the codec packet (see Sect. 4.2.3). Including the redundancy in-band allows the transmission of redundancy to be either channel controlled (e.g., to combat network congestion) or source controlled. In the latter case, the encoder can use properties of the input source signal to determine which frames are most critical for high quality reconstruction at the decoder and selectively transmit redundancy for those frames only. Even better quality can perhaps be gained by including varying degrees of redundancy depending on the criticality as determined from the input.

Another advantage of in-band redundancy is that source control can be used to determine which frames of input can best be coded at a reduced frame rate. In this way, some coded frames can be reduced in size in order to accommodate the attachment of redundancy without altering the total packet size. In this way, redundancy can be incorporated even within a constant bit rate channel. As described in Sect. 4.2.3, the redundancy modes of the forthcoming EVS codec uses this constant bit rate approach.

4.2 Recent Speech Coding Standards

Several standardization organizations have speech coders as their focus. The 3rd Generation Partnership Project (3GPP) and the 3rd Generation Partnership Project 2 (3GPP2) run projects that are especially relevant for mobile communications carried on GSM, UMTS, cdma2000, and LTE radio access networks. The work of the International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) targets wireline and wireless coders while the Internet Engineering Task Force (IETF) focuses on coders for use over the Internet. Traditionally, the Motion Picture Experts Group (MPEG)—a working group of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC)—targeted broadcast standards specifying decoder and bit stream format, but more recently MPEG has conducted activities on conversational coding. In this section, we present the recent developments of speech coder standards for conversational applications (telephony) in these organizations.

4.2.1 Advanced Standards in ITU-T

The International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) has conducted substantial work in standardization of speech and audio codecs. In recent years, ITU-T Study Group 16 standardized enhanced bandwidth codecs like G.722.1 Annex C providing super-wideband capability and G.719 full-band codec. Besides enhancement in audio bandwidth, ITU-T also

pioneered in work toward layered coding. Scalable codecs standardized in ITU-T include G.729.1, G.711.1, G.722, G.722B, G.718, and G.718B. It is common to all ITU-T coders that they are specified by ANSI C source code which takes precedence over the textual description; the ANSI C source code is provided in both fixed-point and floating-point formats.

4.2.1.1 G.729.1: Scalable Extension of G.729

The work to extend the G.729 narrowband coder by bit rate scalability feature started in ITU-T in 2004. The result of this work is the G.729.1 layered codec [33, 45].

G.729.1 provides bit rate and bandwidth scalability at the same time. G.729.1 is the first codec with an embedded scalable structure designed as an extension of an already existing standard, the G.729 coder that is widely used in VoIP infrastructures. Easy integration with existing infrastructure and services required the use of G.729 core codec while a scalable wideband scheme allows simple adjustment of bit rate to network or terminal capabilities and multiplexing in gateways. The extensions add improvement in both quality within narrowband and in audio bandwidth by adding wideband in layers with incrementally increasing quality. In addition to bit rate scalability, the codec also includes a DTX operation which allows encoding of speech at a lower average rate by taking speech inactivity into account.

The layer structure shown in Fig. 4.8 includes three stages with 12 embedded layers. Layers 1 and 2 use the CELP algorithm at 8 and 12 kb/s in narrowband. Layer 2 is bit stream compatible to G.729. Layer 3 works by the Time-Domain Bandwidth Extension (TDBWE) scheme at 14 kb/s in wideband [12]. The algorithm

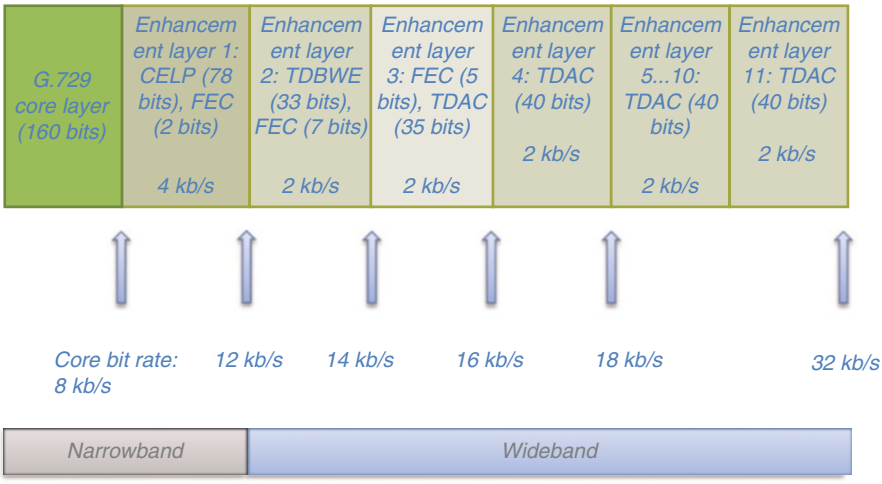

Fig. 4.8 Layered structure of G.729.1 (bits given per 20 ms frame)

in layers 4–12 is predictive transform coding referred to as Time Domain Aliasing Cancellation (TDAC) at 14–32 kb/s in wideband. The bit rates from 14 to 32 kb/s provide an increase in quality.

The G.729.1 codec works on 20 ms frames and has an algorithmic delay of 48.9375 ms. A low-delay mode is also available where the delay is significantly reduced by 25 ms. The worst case complexity is 35.79 WMOPS for encoder + decoder. The CELP algorithm ensures good quality for speech, the TDBWE part allows wideband at reduced bit rate and TDAC provides high quality at wideband for music and non-speech signals. The quality of the 12 kb/s narrowband mode reaches the quality of G.711. Maximum wideband quality for all signal types is achieved at 32 kb/s.

4.2.1.2 G.718: Layered Coder with Interoperable Modes

G.718 [24] is a narrowband and wideband layered coder operating at bit rates between 8 and 32 kb/s. Quality improvements to wideband are offered in 5 layers, and at the highest layers, G.718 Annex B adds super-wideband audio bandwidth.

G.718 works in narrowband at bit rates of 8 and 12 kb/s and in wideband at 8, 12, 16, 24 and 32 kb/s. The codec operates on 20 ms frames and has a maximum algorithmic delay of 42.875 ms for wideband input and wideband output signals. The maximum algorithmic delay for narrowband input and narrowband output signals is 43.875 ms. A low-delay mode is available at 8 and 12 kb/s with a reduced maximum algorithmic delay. Worst case codec complexity is 68 WMOPS.

The coding algorithm is based on a two-stage coding structure: the lower two layers work at 12.8 kHz sampling frequency and are based on CELP coding with signal-classification in the core layer to optimize coding algorithm per frame. The higher layers encode the weighted error signal from the lower layers using overlap-add MDCT transform coding. G.718 achieves a significant performance improvement and contains AMR-WB (G.722.2) interoperable mode as well.

4.2.1.3 Super-Wideband Extensions: G.729.1 Annex E and G.718 Annex B

Super-wideband extensions operate at 32 kHz sampling frequency and implement a scalable coding algorithm to produce a backward compatible embedded bit stream. The bit stream can be truncated at the decoder or by any component of the communication system to instantaneously adjust the bit rate to the desired value with no need for out-of-band signaling.

In case of G.729.1 Annex E [34], the five super-wideband layers result in an overall bit rate range of 36–64 kb/s, building upon the 32 kb/s G.729.1 coder. Bandwidth extension coding is used in MDCT domain of the high-band and enhanced MDCT coding in the low-band. The algorithmic delay is 55.6875 ms.

G.718 Annex B [25] specifies the scalable super-wideband extension using three layers on top of the five layers of G.718 at 32 kb/s, at overall bit rates 36, 40, and 48 kb/s. Optionally, the highest layer (8 kb/s) of G.718 can be omitted which results in the bit rates 28, 32, and 40 kb/s. The super-wideband layers can be applied to the AMR-WB (G.722.2) interoperable bit rates as well. The coding algorithm is similar to G.729.1 Annex E.

4.2.1.4 G.711.1: Scalable Wideband Extension of G.711

G.711.1 applies an embedded scalable structure and provides wideband coding by adding one or two extension layers to the G.711 narrowband core codec working at 64 kb/s [15, 19]. One layer extends the bit rate to 80 kb/s and another layer increases the bit rate further to 96 kb/s. The extensions are designed for low complexity (8.7 WMOPS in worst case) and low delay (11.875 ms with 5 ms frame length). G.711.1 is an embedded coder so the bit stream can be transcoded into G.711 by simple truncation.

The encoder uses a QMF filter bank to form a low-band and a high-band signal. While the low-band is processed by traditional G.711 narrowband core coder at 64 kb/s, the high-band signal passes an MDCT transform and encodes its coefficients at 16 kb/s (first layer), to give a total bit rate of 80 kb/s. A further encoding layer at 16 kb/s is available resulting in a total bit rate of 90 kb/s.

G.711.1 may make benefit of the use of the partial mixing method in conferencing applications. Mixing is performed only partially in the decoded domain such that the core bit streams are decoded and mixed only while the enhancement layers are not decoded, hence the name partial. Instead, one active endpoint is selected from all the endpoints, and its enhancement layers are redistributed to other endpoints as well.

4.2.1.5 Super-Wideband and Stereo Extensions of G.711.1 and G.722

Introduction of stereo feature in speech communication, especially in combination with super-wideband coding, aims at improving user experience further. ITU-T included a layered super-wideband solution and stereo in both G.711.1 and G.722 [27] coders. The super-wideband extension follows the principles of layered coding. There are two types of stereo extensions, one of them is an embedded solution and the other one applies two separate mono coders operating on mid and side signals separately.

As an extension of the G.711.1 wideband coder, Annex D [21] describes a super-wideband and Annex F a stereo scalable layered algorithm. If Annex F stereo is applied on the G.711.1 wideband coder, the resulting bit rates are 96 and 128 kb/s; in case Annex F is applied on top of the super-wideband layered coder (G.711.1 combined with Annex D), five stereo super-wideband bit rates are provided between 112 and 160 kb/s. The coder configurations demonstrate a high

flexibility for applications: mono narrowband (G.711), mono wideband (G.711.1), mono super-wideband (G.711.1 Annex D), stereo wideband (G.711.1 Annex F), and stereo super-wideband (G.711.1 Annex D and F) are all possible with backward compatibility to G.711 and G.711.1, and also mono compatibility in case of stereo.

The G.711.1 Annex F [22] stereo coder uses 5 ms frame length and has an algorithmic delay of 18.125 and 19.0625 ms for wideband and super-wideband, respectively. Both stereo extension layers work at 16 kb/s. While in the first layer basic inter-channel stereo information is transmitted, the second layer transmits inter-channel phase differences of a larger bandwidth, allowing further quality improvement.

The stereo extension of G.722 in Annex D [29] follows the principles of layered coding. The stereo bit rates 64 and 80 kb/s apply to G.722 (wideband) and 80, 96, 112, 128 kb/s apply to G.722 Annex B [28] (super-wideband). Hence, the stereo super-wideband modes are backward compatible with mono wideband and super-wideband, the stereo wideband modes with mono wideband. The stereo coding algorithm is similar to G.711.1.

The layered super-wideband extensions of G.711.1 and G.722 are specified in Annex D and Annex B, respectively. The main characteristics of the coders including bit rates and complexity and delay figures are summarized in Table 4.2. High band enhancement, bandwidth extension and MDCT-based coding are main parts of the extension algorithm.

G.711.1 Appendix IV [23] and G.722 Appendix V [32] describe a coding method for mid-side (MS) stereo while maintaining interoperability with mono transmission. The coding principle is the same in case of G.711.1 and G.722. The left and right input channels are converted into mid and side signals which are then independently encoded by two parallel running G.711.1 coders (Annex D) or G.722-SWB coders (Annex B), respectively. The decoder performs the inverse operation. The LR-MS conversion requires very low complexity.

Table 4.2 Super-wideband extensions in G.711.1 and G.722 coders

Coder	G.711.1		G.722	
Super-wideband specification	Annex D		Annex B	
Frame size (ms)	5		5	
Algorithmic delay (ms)	12.8125		12.3125	
Worst case complexity (WMOPS)	21.498		22.76	
Wideband bit rate (kb/s)	80	96	56	64
Enhanced wideband bit rate (kb/s)	–	–	56	–
Special super-wideband enhancement layer bit rate (kb/s)	–	–	8	–
First super-wideband bit rate (kb/s)	–	–	64	–
First extension layer bit rate (kb/s)	16	16	16	16
Super-wideband bit rate with first extension layer bit rate (kb/s)	96	112	80	80
Second extension layer bit rate (kb/s)	16	16	16	16
Super-wideband bit rate with first and second extension layers (kb/s)	112	128	96	96

4.2.1.6 Full-Band Coding in G.719

The G.719 standard [26] provides low-complexity full-band conversational speech and audio coding in the bit rate range 32–128 kb/s. Full-band coding means the use of 48 kHz sampling frequency and full audio bandwidth ranging from 20 Hz up to 20 kHz. The frame size is 20 ms and the algorithmic delay is 40 ms due to 50 % window overlap. The codec worst-case complexity ranges from 15.4 to 20 WMOPS (increasing with bit rate) where the encoder and decoder use up around the same amount each.

The encoding algorithm is based on transform coding with adaptive time-resolution as it depends on the classification of the actual frame. For stationary frames, MDCT transform is used to obtain a high spectral resolution. For transient frames, a transform providing high temporal resolution is used. Spectral envelope is quantized based on norms of bands obtained from grouping the spectral coefficients and used as input to the adaptive bit-allocation, after adaptive spectral weighting. Low-complexity lattice-vector quantization is applied to the normalized spectral coefficients before encoding. A specialty of the decoder is that spectral components (that were not encoded) are replaced by signal-adaptive noise filling or by bandwidth extension.

4.2.1.7 G.711.0 Lossless Coding

The motivation behind introducing lossless and stateless compression in ITU-T was to address some of the application scenarios of G.711 where the silence suppression must be deactivated. These cases include, for example, high-speed fax or modem, text telephony, and high linearity for effective network echo canceler. The stateless method means compressing the signal per frame independently as the original G.711 decoder can recreate the speech signal on frame basis. On this way, errors cannot propagate from frame to frame. The benefit of lossless coding is that even tandem (multiple consecutive) coding cannot degrade speech quality.

G.711.0 [18] implements lossless and stateless coding at the same time. In VoIP situation, coders are negotiated end-to-end. The lossless and stateless design of G.711.0 allows its use as a compression method on a connection where G.711 has been negotiated, without further signaling or negotiation.

G.711.0 supports both G.711 μ-law and A-law formats and provides frame sizes typically used in IP networks, i.e. 40, 80, 160, 240 and 320 samples. G.711.0 compresses the signal effectively over 50 % in average (as a function of signal type, level, background noise, μ-law or A-law) at low complexity (less than 1.7 WMOPS in worst case) and low memory figure.

G.711.1 Annex C [20] specifies the specific wideband extension when the core codec is the G.711.0 lossless coder.

4.2.1.8 Packet Loss Concealment Algorithms for G.711 and G.722

The purpose of packet loss concealment (PLC) or frame erasure concealment algorithms is to hide packets/frames that were lost due to transmission errors. Since decoding is not possible, the missing portion of the decoded speech signal is replaced based on past history by a synthetic signal which—ideally—makes the loss inaudible. The success of the algorithm depends on the length of the lost segment (short segments can be better filled), on the type of the true speech signal in the lost segment (stationary type is easier to hide) and on the algorithm qualities. Long missing segments likely result in a divergence between true and synthetic signal so muting is inevitable.

In case of G.711, the PLC in Appendix I [17] works in the decoder only. In case of 10 ms frames, a circular history buffer of 48.75 ms is filled with good frames and is used for pitch period calculation and waveform extraction in case of erasures. PLC comes into action when a frame is lost: the pitch is detected first and then during the first 10 ms segment the last 1.25 pitch period is repeated by overlap-add method. For longer erasures, longer pitch period have to be used to obtain necessary quality.

G.722 Appendix III [30] specifies a high-quality PLC algorithm in the decoder. Periodic waveform extrapolation is used to regenerate the lost segments mixing with filtered noise according to good signal characteristics before the loss. The extrapolated signal is split by QMF filter bank and the two sub-band ADPCM encoders are passed to update the states of the sub-band ADPCM decoders. Additional processing improves the quality of the fill. Long missing segments are successively muted.

G.722 Appendix IV [31] describes a low-complexity PLC method where the decoder low-band and high-band signals are conserved at good frames. Erasures are extrapolated in the low-band using linear-predictive coding (LPC), pitch-synchronous period repetition and adaptive muting. In the high-band, the previous frame is repeated pitch-synchronously with adaptive muting and high-pass post-processing. For more details, we refer to Sect. 4.1.5.

4.2.2 IETF Codecs and Transport Protocols

4.2.2.1 Opus Codec

Opus is a speech and audio codec which is capable of operating at a wide range of bit rates starting from 6 kb/s for narrow band mono speech to 510 kb/s for high quality stereo music with algorithmic delay ranging from 5 ms (2.5 ms frame size) to 65.2 ms (60 ms frame size). This flexibility enables different types of applications ranging from conversational speech (real time VoIP) to network music performances or lip sync at live events.

4.2.2.1.1 Core Technologies

Opus combines core technologies from Skype's speech focused SILK codec and Xiph.Org's low latency CELT codec based on the Modified Discrete Cosine Transform (MDCT) to handle music signals. The SILK codec which is based on linear prediction coding has been significantly modified to integrate with Opus and is primarily used to handle narrowband and wideband speech up to ∼32 kb/s. The CELT based core is most efficient on fullband audio (48 kHz sampling rate) and less efficient on low bit rate speech.

Opus also has a hybrid mode which uses SILK and CELT simultaneously for super-wideband and fullband audio bandwidths. In the hybrid mode, the cross over frequency between the two cores is 8 kHz. The SILK layer codes the low frequencies up to 8 kHz by re-sampling the signal to wideband while the CELT (MDCT) layer codes the high frequencies above 8 kHz. The MDCT layer discards the signal below 8 kHz to ensure there is no redundancy in the coding. Opus supports seamless switching between all of its different operating modes.

Audio Bandwidths and Bit Rate Sweet Spots

The Opus codec supports input and output of various audio bandwidths as defined in RFC 6716. The available configurations are summarized in Table 4.3. For a frame size of 20 ms, Table 4.4 shows the bit rate "sweet spots" for the Opus codec:

Variable and Constant Bit Rate Modes of Operation

Opus is inherently designed and is more efficient in the Variable Bit Rate (VBR) mode of operation which is the default. However it also supports a constrained VBR mode which simulates a "bit reservoir" and a true CBR mode without a bit reservoir to impose additional buffering delays. The true CBR mode has lower quality than the VBR modes.

Table 4.3 Opus audio bandwidths and effective sample rates

Abbreviation	Audio bandwidth (kHz)	Effective sample rate (kHz)
NB (narrowband)	4	8
MB (medium-band)	6	12
WB (wideband)	8	16
SWB (super-wideband)	12	24
FB (Fullband)	20	48

Table 4.4 Optimal bit rate ranges for coding different bandwidths with Opus

Bit rate range (kb/s)	Configuration
8–12	Narrowband speech
16–20	Wideband speech
28–40	Fullband speech
48–64	Fullband mono music
64–128	Fullband stereo music

Mono and Stereo Coding

Opus supports both mono and stereo coding within a single stream. The reference encoder tries to make the optimal decision on the number of audio channel (mono or stereo) based on a bit rate versus quality trade off. For example it maybe desirable to encode a stereo input stream in mono since the bit rate maybe too low for sufficient quality. The stereo decoder outputs identical left and right channel upon decoding a mono bit stream and a mono decoder averages the left and right channels upon decoding a stereo bit stream. The number of audio channels can also be specified by the application in real-time.

Packet Loss Resilience

Inter-frame correlation (or prediction) is an important tool to enable good audio quality at low bit rates. However this introduces sensitivity to packet loss (as discussed in Sect. 4.1.5). In Opus, the long term prediction (LTP) filter state is downscaled which in turn reduces the LTP prediction gain only in the first pitch period in the packet. Consequently, the first pitch period has higher residual energy and requires extra bits to encode. The downscaling factor is quantized to one of three values and enables an efficient trade-off between increased bit rate caused by lower LTP prediction gain and improved error resiliency.

Forward Error Correction (Low Bit Rate Redundancy)

Low bit rate encoded versions of perceptually important frames such as voiced onsets or transients are added to subsequent frames to assist in recovery from packet loss. This utilizes the presence of a de-jitter buffer at the receiver for Voice over Packet Switched networks. If the main description of a packet is lost, the decoder can poll the de-jitter buffer to check for future packets carrying low bit rate (i.e., coarser) descriptions of the lost packet for synthesis.

4.2.2.2 RTP Payload Formats

The IETF also specifies packaging formats for codec bit streams transported by the real-time transport protocol (RTP) commonly used by VoIP applications and networks. These so-called *payload formats* sometimes describe multiple packaging formats for each codec, along with a description of information that can be included in a header of the codec payload. Also typically defined in these documents are media types and session description protocol (SDP) parameters and attributes that can be used by clients to negotiate mutually agreeable codec capabilities regarding supported bit-rates, bandwidths, redundancy, and other modes of operation. The details of these formats are outside the scope of this chapter, but the interested reader is encouraged to visit the IETF Datatracker [1], where a simple search for a codec name will reveal the corresponding IETF RFC payload format document.

4.2.3 3GPP and the Enhanced Voice Services (EVS) Codec

3GPP (3rd Generation Partnership Project) is successful in running original development and standardization on the area of speech coding for telephony. Before the creation of 3GPP in 1998, ETSI standardized the GSM full-rate (FR) and half-rate (HR) coders, later the enhanced full-rate (EFR) coder that made calls in GSM possible. These coders are single fix rate coders designed especially for the GSM system. More flexibility was possible by the introduction of the adaptive multi-rate (AMR) coder which operates in narrowband and provides 8 bit rates in the range of 5.9–12.2 kb/s. AMR not only enhances flexibility in GSM through bit rate adaptation but AMR is also the mandatory coder in the 3G (UMTS) system. The introduction of the adaptive multi-rate wideband (AMR-WB) coder [11] improved the quality significantly through the increased audio bandwidth up to 50 Hz to 7 kHz. AMR-WB includes 9 fix bit rates in the range of 6.6–23.85 kb/s where the 12.65 kb/s mode achieves good wideband quality [44]. AMR and AMR-WB were tested for use in packet-switched networks [43].

Enhanced Voice Services (EVS) is the conversational voice codec currently being standardized in 3GPP SA4 for use in next generation voice services primarily over LTE and LTE-A. This work item is slated for completion in Rel-12. When standardized, the EVS codec will be the successor to AMR and AMR-WB coders that are extensively used in 3GPP systems for voice services today. The goals of EVS as envisaged in the TSG-SA TR 22.813 [2] include improved user experience by the introduction of super-wideband (SWB) coding of speech, enhanced quality and system efficiency for narrowband (NB) and wideband (WB) speech services compared to codecs used in pre- Rel-12 voice services, enhanced quality for mixed content and music in conversational applications, robustness to packet loss and delay jitter, and the inclusion of a few modes that are backward interoperable to the current 3GPP AMR-WB codec.

The EVS codec will be the first conversational codec that can encode voice and other audio signals with a super-wideband bandwidth (50 Hz–16 kHz) at bit rates as low as 13.2 kb/s. Super-wideband coded speech sounds closer to the original human voice compared to WB and NB speech and therefore provides a sense of presence. Likewise, for similar bit rates as current 3GPP conversational codecs (AMR and AMR-WB), the EVS codec is expected to offer better quality for NB and WB inputs. Equivalently, the EVS codec is expected to provide improved coding efficiency by coding NB and WB signals at lower bit rates for similar quality as AMR and AMR-WB. The EVS codec is also expected to improve the coding quality for music and mixed content compared to current 3GPP conversational codecs (AMR and AMR-WB) and thus enable improved user experience during in-call music, music on hold etc. The improved robustness to packet loss and delay jitter is expected to lead to optimized behavior in IP application environments like MTSI within the EPS. Further, the bit rates for the EVS coder are selected to optimally utilize the LTE transport block sizes chosen for AMR-WB.

Table 4.5 shows a comparison of features of AMR, AMR-WB and the EVS coders.

Table 4.5 Comparison of AMR, AMR-WB and EVS

Feature	AMR	AMR-WB	EVS
Sampling rates	8 kHz	16 kHz	8 kHz, 16 kHz, 32 kHz, 48 kHz
Audio bandwidth	Narrowband (NB)	Wideband (WB)	NB, WB and Super Wideband (SWB)
Intended input signals	Voice	Voice	Voice & general audio (music, ring tones, & mixed content)
Bit rates (kb/s)	4.75, 5.15, 5.9, 6.70, 7.4, 7.95, 10.2, 12.2	6.6, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85, 23.05, 23.85	5.9, 7.2, 8, 9.6, 13.2, 16.4, 24.4, 32, 48, 64, 96, 128
Mono/Stereo	Mono only	Mono only	Mono and Stereo
Frame size	20 ms	20 ms	20 ms
Algorithmic delay	20 ms / 25 ms	25 ms	Up to 32 ms

4.2.4 Recent Codec Development in 3GPP2

Following on the success of the widely adopted Enhanced Variable Rate Codec (EVRC) for narrowband voice [3], 3GPP2—the standardization body for CDMA2000—standardized EVRC-B in 2007. EVRC-B's source controlled variable rate coding techniques offered network operators the flexibility of multiple capacity operating points to dynamically manage network loads [4], but it was still limited to narrowband coding only.

Also in 2007, 3GPP2 adopted EVRC-WB, which included a split-band wideband coding mode at an average active speech bit rate of 7.42 kb/s along with narrowband modes that are interoperable with EVRC-B [5, 36]. EVRC-WB introduced a highly efficient coding of the high band (3.5–7 kHz) using a linear prediction coding (LPC) scheme combined with non-linear processing of the low band excitation to derive the excitation for high band.

More recently, 3GPP2 combined the WB mode of EVRC-WB with seven NB modes from EVRC-B, resulting in the 2009 standard EVRC-NW (Narrowband-Wideband) [6]. The merging of these codecs gives operators the flexibility to deploy wideband voice services while preserving the ability to dynamically switch to higher capacity narrowband modes to accommodate higher network loads.

In the most recent permutation (completed in 2011), 3GPP2 has introduced EVRC-NW2K—a new service option that replaces one of EVRC-NW's narrowband coding modes with a 2 kb/s coding mode that is intended for use in Extended Cell High Rate Packet Data (xHRPD) systems [8]. One application of such systems is integrated satellite and terrestrial networks that may benefit from significant radio access coverage improvement in exchange for tight bandwidth constraints for transmission of voice over satellite links, while still providing high quality wideband over terrestrial radio links [9]. The operating modes of the five codecs in the EVRC family, including this latest standard, are summarized in Fig. 4.9.

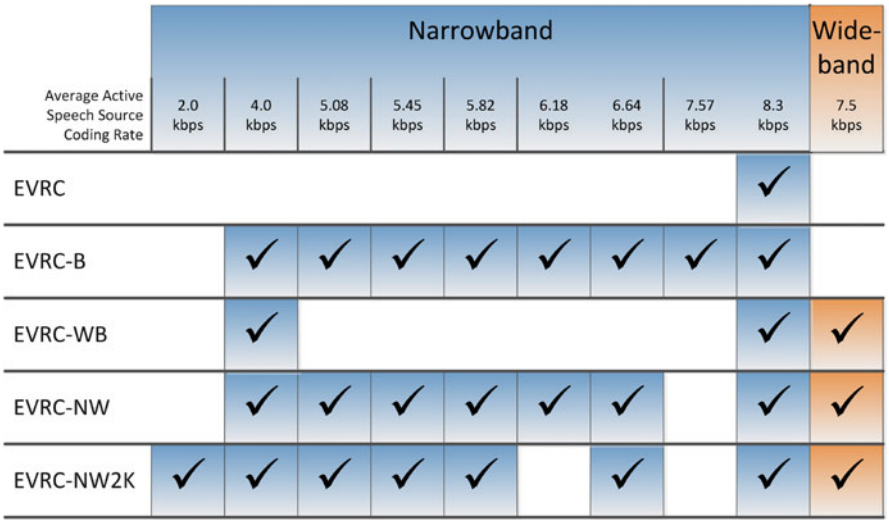

Fig. 4.9 The coding modes of the EVRC family of codecs

The new 2 kb/s coding mode in EVRC-NW2K is achieved using the CDMA rate set one quarter-rate packet (40 bits/packet for source coding) as the maximum packet size. In addition to the noise-excited linear prediction (NELP) coding mode for unvoiced speech from EVRC-B, it also uses a slightly modified version of the quarter-rate prototype pitch prediction (QPPP) from that codec. QPPP is a waveform interpolation based coding for stationary voiced segments. In this mode, prototype pitch periods are extracted from the end of each frame and efficiently encoded using a representation of the discrete Fourier series magnitude spectrum. Phase, which is not encoded with the QPPP prototypes, is extrapolated from the previous frame at the decoder [7]. The whole frame excitation is reconstructed at the decoder by interpolating between the two prototypes.

In addition to NELP and QPPP, EVRC-NW2K's 2 kb/s mode also uses a quarter-rate transient encoding mode. This mode handles voiced transients not well encoded by QPPP as well as plosives, up-transients (typically unvoiced-to-voiced transitions), and down-transients (typically voiced-to-unvoiced transitions). The transient coding also seeds the QPPP mode which requires both a previous pitch prototype (from the previous frame) and phase information [7].

4.2.5 Conversational Codecs in MPEG

ISO/IEC MPEG focused traditionally on broadcast coders, recent activities include the standardization of conversational codecs as well. Low Delay Advanced Audio Coding (AAC-LD) and Enhanced Low Delay Advanced Audio Coding (AAC-ELD) represent recent developments on this area. The AAC-ELD family consists of AAC-LD, AAC-ELD and AAC-ELD v2 [10].

These codecs make use of perceptual audio coding used in advanced broadcast audio coders and it combines with low encoding delay at 15…32 ms (depending on bit rate and bandwidth) that is necessary for conversational applications. Specifically, AAC-LD features a minimum encoding delay of 20 ms at 48 kHz sampling, the three modes of AAC-ELD have 15, 15.7 and 31.3 ms delay, respectively, and AAC-ELD v2 has a typical algorithmic delay of 35 ms. The codec can operate in a fixed frame length mode (20 ms) where each packet is equal in size, or in a fixed bit rate mode where the average bit rate within a limited time frame is constant. AAC-ELD supports various audio bandwidths up to fullband (20 kHz upper bandwidth) and also stereo capability. For stereo, AAC-LD offers natural sound for speech and music at bit rates above 96 kb/s, AAC-ELD improves the audio quality down to 48 kb/s. Below this bit-rate, down to 24 kb/s, AAC-ELDv2 is the best choice to keep the audio quality high. For mono applications, a similar relationship between AAC-ELD and AAC-LD at half bit-rate can be expected, whereas AAC-ELD v2 delivers identical audio quality to AAC-ELD.

The core structure of AAC-LD is directly derived from AAC. The time domain input samples are transformed into a frequency domain representation by an MDCT filter bank. In order to efficiently exploit psychoacoustic effects of frequency and time domain masking, the 960 sample size of the MDCT analysis window utilizes a frequency resolution of 50 Hz and a time resolution of 10 ms. Temporal Noise Shaping allows the AAC-LD coder to exercise control over the temporal fine structure of the audio signal and improve the time resolution. Intensity Coupling and Mid/Side Stereo increase the coding gain for a stereo channel pair compared to encoding two mono channels separately. Perceptual Noise Substitution (PNS) uses a parametric representation of noise-like frequency bands for an efficient transmission.

AAC-ELD can be used at three different operating modes. AAC-ELD core can be used in all applications where high bit rates are available (96 kb/s and higher for stereo). A Low Delay MDCT filter bank replaces the MDCT filter bank used in AAC-LD. With this delay-optimized filter bank, AAC-ELD operates with a lower delay compared to AAC-LD. AAC-ELD with SBR mode is the most flexible mode of AAC-ELD as it covers a wide range of bit rates (approximately 32–64 kb/s per channel) and sampling rates. The delay stays constant over a wide range of bit rates enabling dynamically switching of bit rates. This mode uses a delay-optimized version of Spectral Bandwidth Replication (LD-SBR) technology. LD-SBR allows the reduction of overall bit rate while maintaining excellent audio quality. The lower part of the audio spectrum is coded with AAC-ELD core, while the LD-SBR tool encodes the upper part of the spectrum. LD-SBR is a parametric approach that exploits the harmonic structure of natural audio signals. It uses the relationship of the lower and upper part of the spectrum for a guided recreation of the whole audio spectrum of the signal. The third operating mode, AAC-ELD with Dual Rate SBR is especially useful for applications with lower data rates, down to 24 kb/s per channel, at an increased delay compared to the other two modes. In this mode, the AAC-ELD core is coded with half the sampling frequency of the overall signal

which maximizes quality at low bit rates. Note that AAC-ELD standard-compliant decoders can operate in any of the three modes, which allows the designer of the encoder side to freely choose the mode that best fits the application scenario.

AAC-ELD v2 is ideal for low bit rate stereo operation. It integrates a parametric stereo extension to achieve stereo performance at bit rates close to mono operation. This parametric extension is based on a two-channel version of Low Delay MPEG Surround (LD-MPS). Instead of transmitting two channels, the LD-MPS encoder extracts spatial parameters to enable reconstruction of the stereo signal at the decoder side; the remaining mono down mix is AAC-ELD encoded. The LD-MPS data is transmitted together with the SBR data in the AAC-ELD bit stream. The AAC-ELD decoder reconstructs the mono signal and the LD-MPS decoder recreates the stereo image. Typically, the bit rate overhead for the stereo parameters is around 3 kb/s at 48 kHz. This allows AAC-ELD v2 to code stereo signals at bit rates significantly lower than using discrete stereo coding.

References

1. IETF Datatracker (2014), https://datatracker.ietf.org/doc/
2. Study of use cases and requirements for enhanced voice codecs for the evolved packet system (EPS). 3GPP TSG-SA TR 22.813 (2010), http://www.3gpp.org/ftp/Specs/archive/22_series/22.813/22813a00.zip
3. Enhanced variable rate codec, speech service option 3 for wideband spread spectrum digital systems. 3GPP2 3GPP2 C.S0014-A v1.0 (2004), http://www.3gpp2.org/Public_html/specs/C.S0014-A_v1.0_040426.pdf
4. Enhanced variable rate codec, speech service option 3 and 68 for wideband spread spectrum digital systems. 3GPP2 3GPP2 C.S0014-B v1.0 (2006), http://www.3gpp2.org/Public_html/specs/C.S0014-B_v1.0_060501.pdf
5. Enhanced variable rate codec, speech service options 3, 68, and 70 for wideband spread spectrum digital systems. 3GPP2 C.S0014-C v1.0 (2007), http://www.3gpp2.org/Public_html/specs/C.S0014-C_v1.0_070116.pdf
6. Enhanced variable rate codec, speech service options 3, 68, 70, & 73 for wideband spread spectrum digital systems. 3GPP2 C.S0014-D v3.0 (2010), http://www.3gpp2.org/Public_html/specs/C.S0014-D_v3.0_EVRC.pdf
7. Enhanced variable rate codec, speech service options 3, 68, 70, 73 and 77 for wideband spread spectrum digital systems. 3GPP2 C.S0014-E v1.0 (2011), http://www.3gpp2.org/Public_html/specs/C.S0014-E_v1.0_EVRC_20111231.pdf
8. Introduction to CDMA2000 extended cell high rate packet data air interface specification. 3GPP2 C.S0098-100-0 v1.0 (2011), http://www.3gpp2.org/Public_html/specs/C.S0098-100-0_v1.0_xHRPD_Intro.pdf
9. System requirements for extended cell HRPD (xHRPD). 3GPP2 S.R0143-0 v1.0 (2010), http://www.3gpp2.org/Public_html/specs/S.R0143-0v1.0ExtendedRangexHRPDSRD.pdf
10. The AAC-ELD family for high quality communication services. Fraunhofer IIS Technical Paper (2013), http://www.iis.fraunhofer.de/content/dam/iis/de/dokumente/amm/wp/AAC-ELD-family_TechnicalPaper.pdf
11. B. Bessette, The adaptive multirate wideband speech codec (AMR-WB). IEEE Trans. Speech Audio Process. **10**, 620–636 (2002)

12. B. Geiser, P. Jax, P. Vary, H. Taddei, S. Schandl, M. Gartner, C. Guillaum'e, S. Ragot, Bandwidth extension for hierarchical speech and audio coding in ITU-T Rec. G.729.1. IEEE Trans. Audio Speech Lang. Process. **15**(8), 2496–2509 (2007)
13. B. Geiser, P. Vary, High rate data hiding in ACELP speech codecs, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 (ICASSP 2008)* (2008), pp. 4005–4008. doi:10.1109/ICASSP.2008.4518532
14. M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, SIP: Session Initiation Protocol. RFC 2543 (Proposed Standard) (1999), http://www.ietf.org/rfc/rfc2543.txt. Obsoleted by RFCs 3261, 3262, 3263, 3264, 3265
15. Y. Hiwasaki, H. Ohmuro, ITU-T G.711.1: Extending G.711 to higher-quality wideband speech. IEEE Commun. Mag. **47**(10), 110–116 (2009)
16. ITU-T Recommendation G.114: One-way Transmission Time (2003), http://www.itu.int/rec/T-REC-G.114
17. ITU-T Recommendation G.711 Appendix I: Lower-band postfiltering for R1 mode (2012)
18. ITU-T Recommendation G.711.0: Lossless compression for G.711 PCM (2009)
19. ITU-T Recommendation G.711.1: Wideband embedded extension for ITU-T G.711 (2012)
20. ITU-T Recommendation G.711.1 Annex C: Lossless compression of ITU-T G.711 PCM compatible bitstream in ITU-T G.711.1 (2012)
21. ITU-T Recommendation G.711.1 Annex D: Superwideband extension (2012)
22. ITU-T Recommendation G.711.1 Annex F: Stereo embedded extension for ITU-T G.711.1 (2012)
23. ITU-T Recommendation G.711.1 Appendix IV: Mid-side stereo (2012)
24. ITU-T Recommendation G.718: Frame error robust narrow-band and wideband embedded variable bit-rate coding of speech and audio from 8–32 kbit/s (2008)
25. ITU-T Recommendation G.718 Annex B: Superwideband scalable extension for G.718 (2009)
26. ITU-T Recommendation G.719: Low-complexity full-band audio coding for high-quality conversational applications (2008)
27. ITU-T Recommendation G.722: 7 kHz Audio coding within 64 kb/s (2012)
28. ITU-T Recommendation G.722 Annex B: Superwideband embedded extension for G.722 (2012)
29. ITU-T Recommendation G.722 Annex D: Stereo embedded extension for G.722 (2012)
30. ITU-T Recommendation G.722 Appendix III: A high-quality packet loss concealment algorithm for G.722 (2012)
31. ITU-T Recommendation G.722 Appendix IV: A low-complexity packet loss concealment algorithm for G.722 (2012)
32. ITU-T Recommendation G.722 Appendix V: Mid-side stereo (2012)
33. ITU-T Recommendation G.729.1: G.729 Based embedded variable bit-rate coder: An 8–32 kb/s scalable wideband coder bitstream interoperable with G.729 (2006)
34. ITU-T Recommendation G.729.1 Annex E: Superwideband scalable extension for G.729.1 (2010)
35. ITU-T Recommendation H.323: Packet-based multimedia communications systems (2009), http://www.itu.int/rec/T-REC-H.323
36. V. Krishnan, V. Rajendran, A. Kandhadai, S. Manjunath, EVRC-Wideband: the new 3GPP2 wideband vocoder standard, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007 (ICASSP 2007)*, vol. 2 (2007), pp. II-333–II-336. doi:10.1109/ICASSP.2007.366240
37. Y. Liang, N. Färber, B. Girod, Adaptive playout scheduling and loss concealment for voice communication over IP networks. IEEE Trans. Multimed. **5**(4), 532–543 (2003). doi:http://dx.doi.org/10.1109/TMM.2003.819095 10.1109/TMM.2003.819095
38. M. Dietz, L. Liljeryd, K. Kjorling, O. Kunz, Spectral band replication, a novel approach in audio coding, in *Proceedings of the 112th Convention of the Audio Engineering Society*, vol. 1 (2002)
39. J. Makhoul, M. Berouti, High frequency regeneration in speech coding systems, in *Proceedings of IEEE ICASSP*, vol. 1 (1979)

40. J. Makinen, B. Bessette, S. Bruhn, P. Ojala, R. Salami, A. Taleb, AMR-WB+: a new audio coding standard for 3rd generation mobile audio services, in *Proceedings of IEEE ICASSP*, vol. 2 (2005)
41. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (INTERNET STANDARD) (2003), http://www.ietf.org/rfc/rfc3550.txt. Updated by RFCs 5506, 5761, 6051, 6222, 7022
42. J. Sjoberg, M. Westerlund, A. Lakaniemi, Q. Xie, RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs. RFC 4867 (Proposed Standard) (2007), http://www.ietf.org/rfc/rfc4867.txt
43. H. Taddei, I. Varga, L. Gros, C. Quinquis, J.Y., Monfort, F. Mertz, T. Clevorn, Evaluation of AMR-NB and AMR-WB in packet switched conversational communications, in *International Conference on Multimedia and Expo (ICME)* (2004)
44. I. Varga, R.D.D. Iacovo, P. Usai, Standardization of the AMR wideband speech codec in 3GPP and ITU-T. IEEE Commun. Mag. **44**(5), 66–73 (2006)
45. I. Varga, S. Proust, H. Taddei, ITU-T G.729.1 scalable codec for new wideband services. IEEE Commun. Mag. **47**(10), 131–137 (2009)
46. S. Voran, Subjective ratings of instantaneous and gradual transitions from narrowband to wideband active speech, in *Proceedings of IEEE ICASSP* (2010)
47. M. Yavuz, S. Diaz, R. Kapoor, M. Grob, P. Black, Y. Tokgoz, C. Lott, VoIP over cdma2000 1xEV-DO revision A. IEEE Commun. Mag. **44**(2), 88–95 (2006)

Part II
Review and Challenges in Speech, Speaker and Emotion Recognition

Chapter 5
Ensemble Learning Approaches in Speech Recognition

Yunxin Zhao, Jian Xue, and Xin Chen

Abstract An overview is made on the ensemble learning efforts that have emerged in automatic speech recognition in recent years. The approaches that are based on different machine learning techniques and target various levels and components of speech recognition are described, and their effectiveness is discussed in terms of the direct performance measure of word error rate and the indirect measures of classification margin, diversity, as well as bias and variance. In addition, methods on reducing storage and computation costs of ensemble models for practical deployments of speech recognition systems are discussed. Ensemble learning for speech recognition has been largely fruitful, and it is expected to continue progress along with the advances in machine learning, speech and language modeling, as well as computing technology.

5.1 Introduction

Speech recognition is a challenging task. Producing a spoken message requires conceptualizing what to say based on a semantic memory, formulating words and their ordering according to a language syntax, and articulating the message following a phonetic and articulatory planning. Speech data that are produced from the multi-tiered process are not i.i.d., the temporal dynamics of speech sounds and their spectral energy distributions are doubly stochastic and subject to rich variations in speaking style, speech rate, speaker differences, acoustic background, etc. It is obviously difficult to learn any single model in any component of the system to fully accommodate the structures and variations of such complex speech data, and it

Y. Zhao (✉)
Department of Computer Science, University of Missouri, Columbia, MO 65211, USA
e-mail: zhaoy@missouri.edu

J. Xue
Microsoft Corporation, Bellevue, WA 98004, USA
e-mail: jianxue@microsoft.com

X. Chen
Pearson Knowledge Technology Group, Menlo Park, CA 94025, USA
e-mail: xin.chen@pearson.com

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_5

is more so when the amount of supervised speech training data is often very limited for an application task. These difficulties make ensemble learning very attractive for speech recognition, as it offers a conceptually simple framework to synthesize a more accurate and robust learner from some simple learners that are trainable from limited data. The hierarchical nature of speech also allows ensemble learning for speech recognition to appear in different system components and at different levels of decision making.

Over the years, many efforts on ensemble learning have emerged in the speech recognition field. The reported methods share certain commonalities with those in machine learning, mostly with the ensemble generation mechanisms of boosting, bagging, and random forest, but they invariably carry special traits pertaining to their choices on the system components where ensemble learning is performed and on the system level where diversity integration is applied. Generally speaking, diversity integration has appeared in every possible level of decoding search; while the ensemble learning methods are mainly focused on acoustic modeling, they are also applied to language modeling and speech features, and in some cases ensemble acoustic and language models are combined.

In this chapter, an overview is made on the ensemble learning approaches that have emerged in automatic speech recognition in recent years in connection with those of the machine learning literature. In Sect. 5.2, a background is given to the ensemble learning methods in machine learning with the focus on classification. In Sect. 5.3, the key concepts and components of state-of-the-art speech recognition systems are described. In Sect. 5.4, ensemble learning for speech recognition is discussed in terms of opportunities of injecting diversity or randomness into different components of a speech recognition system as well as possibilities for combining multiple models or systems at different levels of decision making. In Sect. 5.5, ensemble learning methods in acoustic modeling is categorized and examined in detail. In Sect. 5.6, a light discussion is given to some ensemble methods in language modeling. In Sect. 5.7, performance enhancing mechanisms of ensemble methods in acoustic modeling are analyzed. In Sect. 5.8, methods for streamlining ensemble acoustic models to improve computation and storage efficiencies are discussed. In Sect. 5.9, a conclusion is made.

5.2 Background of Ensemble Methods in Machine Learning

5.2.1 Ensemble Learning

Ensemble learning has become an important and active research area over the past decade, covering the full spectrum of supervised learning for classification and regression, unsupervised learning for data clustering, and semi-supervised learning [25, 50, 84]. For classification, an ensemble learner or classifier builds a set of classifiers and combines their predictions for each test sample. A convenient

assumption that has commonly been made is that the data samples are independent, and thus the classification deals with isolated objects, which simplifies problem formulation and analysis. Numerous studies on a variety of machine learning tasks have provided ample empirical evidences that the predictions made by combining multiple classifiers are often more accurate than the predictions made by the individual classifiers.

Learning an ensemble classifier encompasses learning the component classifiers, often referred to as base classifiers, and their combining weights. An ensemble classifier may be heterogeneous or homogeneous, where in the former the base classifiers are of mixed types, for example, a decision tree, an artificial neural network, and a k-nearest neighbor classifier, while in the latter the base classifiers are of the same type, such as all being decision trees. Homogeneous ensemble construction methods of boosting and bagging have been heavily studied in the machine learning community.

5.2.2 Boosting

Boosting is a sequential method for ensemble construction, where the base classifiers are learnt one after another from reweighted data. Specifically, the base classifiers that have been learnt at the current stage are combined to classify the current version of weighted training data, and the prediction errors are utilized to reweight the data distribution so as to emphasize the misclassified data samples in learning the next base classifier. Among a variety of boosting algorithms, AdaBoost is the most influential [36], which has been shown to increase classification margins on training data and thus decrease generalization errors on unseen data with the boosting iterations. Schapire [65] established the notion that weak learners that are just slightly better than random guess can be combined into a strong learner through boosting. A statistical estimation based analysis revealed that Adaboost is equivalent to forward stage-wise additive modeling [39].

Boosting algorithms can be categorized by the number of classes that it discriminates being binary or multiple, by its output being discrete or real, and by the loss function that it employs in the ensemble optimization. In the discrete case, such as AdaBoost, the outputs are class labels alone, while in the real case, such as Real AdaBoost, the outputs provide information of the class labels as well as the confidence scores of the class predictions [34]. A variety of loss functions have been investigated that has led to variants of boosting algorithms. The exponential loss as employed in AdaBoost increases the weights of the misclassified samples exponentially with their negative classification margin sizes, making AdaBoost sensitive to noise [39]. To decrease noise sensitivity, the loss function can be made to increase with the negative margin size at a smaller rate, such as the Binomial deviance loss in FilterBoost [3].

5.2.3 Bagging

Bagging is a parallel method of ensemble construction, where the base classifiers are learnt independently from bootstrap-sampled datasets. For a training set D with N samples, bootstrap sampling generates K sampled datasets D_i, $i = 1, \cdots, K$, by randomly sampling D with replacement N times such that $|D_i| = D$. K base classifiers are learnt from the D_i's independently, and their predictions on unseen data are aggregated to form the ensemble prediction [4]. Because each D_i would randomly miss on average 36.8 % of samples in D, the base classifiers are different and potentially complementary to each other in decision making. In order for the limited differences among the D_i's to generate large differences among the base classifiers, the classifiers should be sensitive to small changes in training data. Friedman and Hall [33] analyzed the effects of bagging on the parameters of ensemble classifiers and showed that linear parameters were not improved while the variance and higher-order terms of the parameters were reduced, implying that nonlinear classifiers were good candidates for bagging. The performance of bagging improves with the number of base classifiers or the ensemble size generally, and it converges eventually [84]. Subagging is similar to bagging but it uses subsampling (random sampling without replacement) to generate sampled datasets of a smaller size than the given dataset [2].

5.2.4 Random Forest

Since classification trees are unstable, they are commonly used in bagging. A generalization in randomization has led to the celebrated ensemble classifier called Random Forest. Random Forest as described in Breiman [5] includes two aspects of randomization in training the base classifiers: randomizing data through bootstrap sampling, and randomizing features of node splits by randomly selecting a subset of features for each node. Random forest covers a variety of ensemble methods that use random tree base classifiers. For example, the method of Random Subspace randomly chooses a subset of features for each base tree construction without randomizing node splits [41]; Randomized C4.5 randomly chooses a node split from n-best split candidates for each node of a base C4.5 tree [24]; Rotation Forest employs different feature rotations that are derived from principal component analyses on randomly sampled data to randomize the base tree classifiers [63]. It is worth mentioning that the last three methods all use the full training set D to train the base tree classifiers, and the randomness of these trees are derived from the feature manipulations.

5.2.5 Classifier Combination

How to combine the base classifiers to achieve accurate ensemble predictions is an important issue. Existing approaches include static classifier combination, dynamic classifier selection, stacking, etc.

In static classifier combination, the combining weights for the base classifiers are determined in the training stage and held fixed. Assuming that C classes are to be discriminated, the output of the ith classifier can be represented by the vector

$$h_i(x) = \left(h_i^1(x)\ h_i^2(x) \cdots h_i^C(x) \right)^T \tag{5.1}$$

where T stands for vector transpose. If the classifier produces class labels, then $h_i(x)$ is in the form of 1-of-C code. For example, $h_i^j(x) = 1$ and $h_i^k(x) = 0$ for $k \neq j$, when the class of x is predicted to be j. If the outputs are class scores, then $h_i(x)$ is a real vector. For example, for an artificial neural network, $h_i(x)$ is a vector of class posterior probability estimates. For an ensemble with K base classifiers, its output vector becomes

$$H(x) = \left(\frac{1}{K}\sum_{i=1}^{K} \alpha_i h_i^1(x)\ \frac{1}{K}\sum_{i=1}^{K} \alpha_i h_i^2(x) \cdots \frac{1}{K}\sum_{i=1}^{K} \alpha_i h_i^C(x) \right)^T \tag{5.2}$$

where the α_i's are combining weights, and the ensemble prediction is commonly

$$j^* = \arg\max_{1 \le j \le C} \sum_{i=1}^{K} \alpha_i h_i^j(x) \tag{5.3}$$

In a simple classifier combination, the weights are uniform and the decision rule is majority voting. Bagging falls into this case. Otherwise, the combining weights are estimated, for example, by making the elements $\frac{1}{K}\sum_{i=1}^{K} \alpha_i h_i^j(x)$ approximate the class posterior probabilities $p(j|x)$. It is worth noting that the combining weights in AdaBoost are exponential functions of the error rate of the base classifiers, which are learnt sequentially with the base classifiers to minimize classification errors of the ensemble classifier.

In dynamic classifier selection, one or a subset of base classifiers are selected for predicting the class of a sample x, with the objective of including the classifiers that are likely to be correct for x and avoiding those that are likely to be wrong. To do so, the accuracies of the base classifiers on individual training samples are tracked, and the base classifiers that perform well for the training samples that are similar to x are selected.

In stacking, the outputs of the base classifiers on training samples are used as features to train a meta-classifier [76]. The meta-classifier is nonlinear in general,

and thus stacking provides a more general way for classifier combination. To avoid overfit, cross-validation is usually used to separate the training data into those for the base classifiers and those for the meta-classifier. Once the meta-classifier is learnt, the base classifies can be re-trained by the full set of training data. Beside of this commonly adopted two-level stacking, classifiers can also be stacked on top of each other to form a vertical structure of multiple layers [76], offering a way for deep learning.

5.2.6 Ensemble Error Analyses

The error reduction mechanism of an ensemble learner over its base learners has been analyzed for classification and regression, and the consensus is that the diversity among the base learners contributes to the performance gain of an ensemble learner.

5.2.6.1 Added Error of an Ensemble Classifier

Tumer and Ghosh [72] analyzed the classification errors that are added to the intrinsic Bayes error for the ensemble learner that uses a simple soft voting, and they derived the relation between the expected added error of the ensemble learner $E_{add}(H)$ and the average expected added error of K unbiased base learners $\overline{E}_{add}(h)$:

$$E_{add}(H) = \frac{1 + (K-1)\,\delta}{K}\overline{E}_{add}(h) \tag{5.4}$$

where δ is the average pair-wise correlations among the base leaners. This relation shows that reducing the correlations δ among the base leaners reduces the added error of the ensemble, and when the base learners are uncorrelated, i.e., $\delta = 0$, $E_{add}(H)$ becomes a factor of K smaller than $\overline{E}_{add}(h)$. It is also seen from Eq. (5.4) that accurate base classifiers and large ensemble size both contribute to a good ensemble performance.

5.2.6.2 Bias–Variance–Covariance Decomposition

In regression, a learner h is trained from a training set D to approximate a target function f. The squared approximation error can be decomposed as a bias term plus a variance term when taking an expectation with respect to the probability distribution of D:

$$E\left[(h-f)^2\right] = bias(h)^2 + \mathrm{var}(h) \tag{5.5}$$

with $bias(h)^2 = (E[h] - f)^2$ and $\mathrm{var}(h) = E[(h - E[h])^2]$.

For an ensemble of K base learners using a simple averaging rule, i.e., $H = \frac{1}{K}\sum_{i=1}^{K} h_i$, its approximation error can be decomposed as a bias term plus a variance term plus a covariance term [74]:

$$E\left[(H-f)^2\right] = \overline{bias}(H)^2 + \frac{1}{K}\overline{\text{var}}(H) + \left(1-\frac{1}{K}\right)\overline{\text{cov}}(H) \tag{5.6}$$

where $\overline{bias}(H) = \frac{1}{K}\sum_{i=1}^{K} bias\,(h_i)$, $\overline{\text{var}}(H) = \frac{1}{K}\sum_{i=1}^{K} \text{var}\,(h_i)$, and $\overline{\text{cov}}(H) = \frac{1}{K(K-1)}\sum_{i=1}^{K}\sum_{j=1, j\neq i}^{K} \text{cov}\left(h_i, h_j\right)$. This decomposition reveals that the bias term of the ensemble learner is the average bias of its base learners, the variance term of the ensemble learner is a factor of K less than the average variance of its base learners. The opportunity of error reduction for the ensemble learner comes from the reduced variance, as well as negative covariances among the base learners.

5.2.6.3 Error-Ambiguity Decomposition

For an ensemble regression learner using a soft averaging, i.e., $H(x) = \sum_{i=1}^{K} \alpha_i h_i(x)$, with $\alpha_i \geq 0$ and $\sum_{i=1}^{K} \alpha_i = 1$, its approximation error can also be decomposed into the two terms of error and ambiguity [48]:

$$\begin{aligned} E_X\left[(H(x)-f(x))^2\right] = &\sum_{i=1}^{K} \alpha_i E_X\left[(h_i(x)-f(x))^2\right] \\ &- \sum_{i=1}^{K} \alpha_i E_X\left[(h_i(x)-H(x))^2\right] \end{aligned} \tag{5.7}$$

where the first term is the expected squared error of the base learners, and the second term is the expected squared difference of the base learners and the ensemble, called ambiguity. The non-negativity of the ambiguity term indicates that the ensemble error is no larger than the average error of the base learners.

5.2.7 Diversity Measures

In binary classification, several diversity measures have been derived from the pair-wise agreement and disagreement patterns of the base classifiers' outputs, including Kappa-statistic, correlation coefficient, disagreement measure, etc.; diversity can also be measured directly from the correct and incorrect classification counts of the base classifiers as in Kohavi–Wolpert variance and Interrater agreement, or from the base classifiers' output class counts on each sample as in Entropy [50].

Information-theoretic diversity measures have been proposed recently to decompose the mutual information between an ensemble classifier's predictions and the ground truth classes into a relevance term and a diversity term [8]. The relevance term is a sum of mutual information between the individual classifier's predictions and target classes, which measures the accuracy of the base classifiers. The diversity term is the conditional mutual information among the base classifiers given target class labels minus the unconditional mutual information among the base classifiers. The mutual information diversity measures are attractive, but computing the diversity term requires the joint probability distributions of the base classifiers which are difficult to estimate for large ensembles and large number of classes. Zhou and Li [83] described a simpler formulation and an approximation method for information-theoretic diversity measure.

Although diversity is important to the accuracy performance of an ensemble classifier, maximizing a diversity objective explicitly in constructing an ensemble classifier showed mixed results [49]. Tang et al. [70] discussed deficiencies in the existing diversity measures, listing non-monotonic relations between diversity and minimum classification margin of an ensemble, lacking of regularization in diversity based objectives which could cause overfit, and correlations between the accuracy of the base classifiers and the diversity among them. Overall, finding the right diversity objectives for ensemble construction remains an open problem at the current time.

5.2.8 Ensemble Pruning

A large ensemble classifier may take up a lot more memory space and computation time than a single classifier does if its base classifiers are not simple. It may be of practical interest to trim an ensemble by keeping only the accurate and diverse base classifiers and prune away the rest of them. In an ensemble classifier whose combining weights carry the importance of its base classifiers, the base classifiers with insignificant weights can be pruned. The base classifiers can also be clustered and only the representative classifiers of the different clusters are retained for the pruned ensemble classifier. Along this line, a base classifier can be represented by a vector whose elements are its classification output labels on a training samples as in Eq. (5.1), and K-means clustering can then be performed on the vectors of all base classifiers and training samples to generate the representative classifiers [51]. The base classifiers can also be ranked by their accuracy or pair-wise

diversity, and those ranked at the top are included in the pruned ensemble, such as in Reduced error pruning and Kappa pruning [54]; or alternatively, a pruned ensemble can be initialized with the best base classifier, and additional base classifiers are added to it sequentially to maximize the diversity between the new base classifier and the pruned ensemble classier, as in Inter-rater agreement pruning and Complementariness pruning [55].

5.2.9 Ensemble Clustering

Since commonly used clustering methods are sensitive to a number of factors, such as initial condition, distance measure, hyper-parameters, clustering algorithm, or data samples, different base clusterings can be readily generated by varying these factors. To reconcile different clusterings into an ensemble clustering, a variety of methods can be used, such as performing average-linkage agglomerative clustering via an average similarity matrix of the base clusterings [32], or representing base clusterings in a graph and performing graph partition [69]. Overall, an ensemble clustering captures multifaceted data associations better than a single clustering does, and the data structure discovered from an ensemble clustering is more robust than that from a single clustering.

5.3 Background of Speech Recognition

5.3.1 State-of-the-Art Speech Recognition System Architecture

A block diagram of a generic state-of-the-art large vocabulary continuous speech recognition (LVCSR) system is shown in Fig. 5.1. The system consists of two function modules: front-end processing and decoding search, and three knowledge

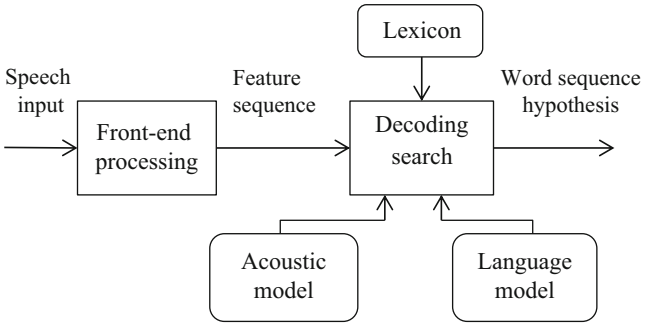

Fig. 5.1 Block diagram of a generic speech recognition system

sources: acoustic model, language model, and lexicon [45, 61]. The front-end processing module processes a microphone output to extract speech from background or segment speech of different talkers, as well as to perform feature analysis to generate a sequence of speech feature vectors $O = o_1 o_2 \ldots o_T$ which may be further transformed for normalization or dimension reduction. The decoding search module utilizes the system knowledge to optimize the word sequence hypothesis W^* for the acoustic evidence O by maximizing the posterior probability of W, i.e.,

$$\begin{aligned} W^* &= \arg\max_W P\left(W\middle|O\right) \\ &= \arg\max_W P(W) P\left(O\middle|W\right) \end{aligned} \tag{5.8}$$

The word sequence probability $P(W) = \prod_{i=1}^{n} P(w_i|w_1, \cdots, w_{i-1})$ is computed by the language model with w_0 a fixed sentence-start symbol. The likelihood of a speech observation sequence is commonly approximated as $P(O|W) \approx \max_{s_1 s_2 \cdots s_T} \prod_{t=1}^{T} p(o_t|s_t) P(s_t|s_{t-1})$, with the state sequence $s_1 s_2 \cdots s_T$ determined by the Viterbi algorithm according to word prediction probabilities, word pronunciations, and topology of subword unit HMMs, and the frame likelihood scores $p(o_t|s_t)$ and the state transition probabilities $P(s_t|s_{t-1})$ computed by the acoustic model.

5.3.2 Front-End Processing

The major task of this module is extracting features from speech. Speech features are commonly analyzed from short-time frames on the scale of 20–30 ms per frame at the rate of about 100 frames per second. A feature vector o_t represents the speech spectral characteristics of the t-th frame. Generally, a discrete Fourier spectrum of a frame is subject to the mel or critical band frequency warping as well as the log or cubic root energy compression to emulate the frequency and energy sensitivity of the human ear. A variety of feature representations exist, and two commonly used ones are Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction Cepstral Coefficients (PLP-CC). To capture speech temporal dynamics in the features, for each frame, some difference of the adjacent frames' speech spectra, referred to as the Δ feature, and difference of the differences, referred to as the $\Delta\Delta$ feature, are used with the current speech spectrum to form a feature vector for the frame. Alternatively, for each frame, speech spectra of several adjacent frames can be spliced as a long vector, and discriminative or orthogonal transformation based dimension reduction are applied to produce a feature vector. Another type of speech feature, referred to as tandem feature, consists of multiple-layer perceptron (MLP) output posterior probabilities of subword units such as phone units. The MLP

is trained to minimize subword classification errors by taking a block of spectral feature vectors as input at each time.

5.3.3 Lexicon

A simple lexicon defines the pronunciation of each vocabulary word by a sequence of phonemes. Because word pronunciations vary with dialect accents, speaker idiosyncrasies, and heteronyms, etc., a lexicon may use multiple pronunciations for some words, with the frequency of each pronunciation variant represented by a weight. While phoneme based lexicons have been widely used in conjunction with acoustic modeling of context-dependent phone units, the approach is often considered too rigid for co-articulations and reductions in spontaneous speech. There are exploratory efforts on modeling word pronunciations by asynchronous streams of multiple articulatory features [18, 52], as well as using longer subword units such as demi-syllables [77].

5.3.4 Acoustic Model

For several decades, the dominant approach to speech acoustic modeling has been hidden Markov modeling (HMM) of context-dependent phone units. In the commonly used triphone models, the context of a phone unit includes its immediately preceding phone and following phone. An HMM describes the stochastic properties of a phonetic sound in two layers: a hidden layer that describes the time dynamics of the feature vectors through a finite-state Markov chain, and an observation layer that describes the statistical variations of the observed feature vectors through state-dependent probability density functions (pdf) that are commonly Gaussian mixture models (GMM):

$$f_s(o_t) = \sum_{i=1}^{I} c_{s,i} N(o_t; \mu_{s,i}, \Sigma_{s,i}) \tag{5.9}$$

where s denotes a state, I denotes mixture size, with the mixture weights $c_{s,i} \geq 0$ and $\sum_{i=1}^{I} c_{s,i} = 1$.

To facilitate training triphone HMMs, the feature sequence of each training speech utterance is first segmented into phone states via a Viterbi alignment. Viterbi alignment maximizes the posterior probability of the phone state sequence given

a feature sequence by using an initial acoustic model and the word transcript of the utterance as well as the pronunciations of these words. Since there are many triphones while the amount of training data is often limited for a given task, the triphone states are clustered into a smaller number of tied triphone states to facilitate reliable parameter estimation for the state-dependent observation pdfs.

Phonetic decision trees (PDT), like those implemented in the HTK toolkit [42], are commonly used for triphone state clustering. For each phone unit P and state s, a PDT is constructed to cluster the triphone samples $P_L - P + P_R$ at the state s. A set of questions, $Q = \{q_1, \cdots, q_L\}$, is predefined for node splits, with each question concerning a certain property of the left and right phone neighbors P_L and P_R. At each node, a Gaussian density is used to fit the data, questions are asked to try out all tentative two-way splits of the triphone samples, and the split that results in the largest likelihood gain is taken. Starting from the root node, node split proceeds iteratively, where a node becomes a leaf when its sample count or the likelihood gain from a further split falls below predefined thresholds. For each leaf node, a GMM is estimated from the triphone samples clustered in the node by the Expectation-Maximization (EM) algorithm. After the PDT-based state tying, the GMM–HMM parameters of the tied-states are further refined by Baum–Welch based maximum likelihood estimation (MLE) to optimize model-data fit, or by discriminative training (DT) to minimize word or phone error rates, where in either case several iterations are run over the feature sequences of training speech.

Another approach to speech acoustic modeling is to use the phone state posterior probabilities of MLP in place of the GMM likelihood scores in HMM, referred to as MLP–HMM hybrid. In recent years impressive progresses have been made in this direction by using deep neural network produced posterior probabilities of context-dependent phone states in MLP–HMM hybrid, referred to as DNN–HMM. DNN–HMM has enjoyed large word error reductions over GMM–HMM in some tasks [17], and deep learning has become an important direction for acoustic modeling.

5.3.5 Language Model

The prevalent approach to language modeling for speech recognition has been the statistical n-gram model, where the prediction of the ith word given the past $i-1$ words, i.e., $P(w_i|w_1 \cdots w_{i-1})$, is approximated by $P(w_i|w_{i-n+1} \cdots w_{i-1})$, that is, different word histories with the same recent $n-1$ words are considered equivalent for predicting the next word. While n-gram language models effectively characterize the short-range lexical co-occurrence frequencies in a language, it is deficient in capturing long-distance word dependencies that may occur in sentences of complex syntax structures. To model more complex language phenomena, syntactic and semantic language models have been developed for speech recognition in combination with n-gram models. Coupled with the progresses in deep learning,

recurrent neural network (RNN) based language modeling which provides smooth word-prediction probabilities and uses long word histories has shown a promising potential for speech recognition [58].

5.3.6 Decoding Search

In large vocabulary continuous speech recognition, searching for the optimal word sequence hypothesis W^* requires intensive computation and huge memory space, and thus fast and memory efficient decoding search algorithms are needed. One commonly used algorithm is one-pass time-synchronous Viterbi beam search based on a lexical prefix tree. The search expands words, phonemes, and HMM states while covering the input speech feature sequence from left to right. The log language and acoustic model scores of different search paths, $\log p(w_1 \cdots w_i)$ and $\log p(o_1 \cdots o_t | w_1 \cdots w_i)$, are accumulated during the search path expansions, and upon reaching the end of the speech input, the path with the largest $\log P(W, O)$ is backtracked to give the 1-best word sequence hypothesis.

In order to apply better models to improve word accuracy without significantly slowing down decoding search, a two-pass search strategy called lattice rescoring is often adopted. In this approach, simple or moderate acoustic and language models are used to generate a word lattice consisting of weighted word arcs in a directed acyclic graph, where the weights are the acoustic and language model scores of the associated words. Since the search space in a word lattice is much smaller than that in the first-pass search, complex acoustic or language models can be used to refine the scores on the word arcs, and word sequence hypothesis can be improved by searching for the best path on the rescored lattice.

Another search strategy is to convert a word lattice into a linear graph called confusion network (CN) via a time-constrained clustering of the word arcs in a word lattice. A CN consists of a sequence of bins, where each bin has a set of aligned words, and each word has its posterior probability of occurrence in the bin that is computed from the word lattice by using a forward–backward algorithm. The best word sequence hypothesis is obtained from a CN by picking from each bin the word candidate with the highest posterior probability. An advantage of the CN is attributed to its direct minimization on word error rate, in contrast to Viterbi search or lattice rescoring that minimize word string error rate.

Unlike the dynamic search space expansion in Viterbi time-synchronous beam search, weighted finite state transducer (WFST) precompiles system knowledge of language model, lexicon, and acoustic model into a large network. Generally speaking, WFST gives faster decoding speed but uses more memory space than the one-pass Viterbi search does, and WFST has become increasingly used in speech recognition systems.

5.4 Generating and Combining Diversity in Speech Recognition

5.4.1 System Places for Generating Diversity

There are ample opportunities for injecting diversities or randomness in a speech recognition system. Along this line, researchers have devised novel methods for almost every component of the speech recognition system depicted in Fig. 5.1. Likewise, the diversity, for example multiple models, can also be exploited at almost every level of speech recognition. Here, the various options that have been explored are synopsized, with the emphasis on acoustic modeling.

5.4.1.1 Front End Processing

Different types of feature analyses can be performed to extract complementary information from a speech signal. These include short-time spectral features such as MFCC, PLP, and filter bank, class posterior probability features from MLP for phone-states or articulatory descriptors; long-time features such as RASTA-PLP, modulation spectrum, and TRAP, which integrate information at the speech syllable rate and are suitable for longer subword units and robust to reverberation. There are also systematic manipulations on time and frequency scales to generate multi-resolution spectral features. Different transformations can also be applied to the same or different types of features to create diversity, for example, Gaussianization, linear discriminative analysis, heterogeneous linear discriminative analysis (HLDA), etc. The multiple feature representations can be treated as parallel feature streams or they can be concatenated into a long vector for each frame as one feature stream. In addition, automatically segmenting an audio input into speech segments of different speakers is often a part of front-end processing. As it is difficult for any algorithm to produce a perfect segmentation, multiple algorithms are used to produce multiple segmentations, and on top of which multiple speech hypotheses can be generated and combined.

5.4.1.2 Acoustic Model

There is a rich collection of methods for producing diversity in acoustic models. Different model training criteria are available, with the major dichotomy lying between maximum likelihood estimation and a variety of discriminative training based estimation. To generate multiple datasets, speech data are resampled randomly as in bootstrap sampling and subsampling, or deliberately as in boosting or clustering, and from which multiple acoustic models are trained. For models that use PDT-based state tying, randomizing PDTs leads to randomized acoustic models. For each phone-state, the randomized PDTs form a random forest (RF)

for multi-way phonetic context clustering, which is different from the classification or regression RFs in the machine learning literature. Using deterministic PDTs, multiple acoustic models can also be created by generating multiple GMMs for each tied state. Multiple GMM–HMM acoustic models can be further produced by using different mixture sizes in different models, or by varying the covariance matrix structures of Gaussian pdfs, such as using diagonal covariance matrices in one model, and full covariance matrices in another model. In MLP–HMM hybrid, an MLP ensemble can be trained to produce multiple outputs of class posterior probability scores for each frame.

It is worth mentioning that the subword units chosen for acoustic modeling interact with the way that word pronunciations are described in a lexicon. Phone units are by far the most widely used in lexicons and acoustic models, but asynchronous articulatory states and syllable-like units are also explored and they can be combined with the phone based models. In such a scenario, lexicon and acoustic model can be viewed as contributing diversity jointly to a recognition system.

5.4.1.3 Language Model

An important problem in language modeling is to capture word dependencies of sufficient ranges without running into the sparse data problem. There is a rich literature in smoothing word prediction probabilities through combining language models of different history lengths. One commonly adopted approach is to smooth the n-gram language model probabilities by backing off an n-gram with insufficient count to an (n–1)-gram with sufficient count, notably the modified Kneser-Ney back-off method [9]. There are also efforts on integrating language models with different focuses on lexical, syntactic, and semantic aspects of a language. While these approaches are under the general umbrella of multiple model combinations, they are beyond the scope of this chapter. On the other hand, ensemble language models have also been generated by using some typical machine learning methods such as random forest, data sampling, parameter and structure randomization, etc. In Sects. 5.5 and 5.6, light discussions are given on three such methods to complement the heavier discussions on ensemble learning for acoustic models.

5.4.2 System Levels for Utilizing Diversity

Speech recognition as formulated in Eq. (5.8) is a sequence optimization problem. Although a sequence can be treated as an entity for classification by an ensemble system, its lower level constituents of words, phonemes, states, or features all present chances for combining the base components of an ensemble system to improve sequence decision accuracy, and even multiple speech inputs can be combined as in a microphone array to improve speech quality for noise-robust

speech recognition, but the latter case is beyond the scope of this chapter. Since speech recognition performance is commonly measured by word error rate, the lower level approaches may yield results that are more relevant to the performance measure. For real-time systems, another important issue is the way that the system level of utilizing diversity impacts decoding search.

5.4.2.1 Utterance Level Combination

Generally, for utterance level combination, decoding search is run multiple times on a speech utterance to generate multiple word sequence hypotheses, where in each run a different base component of an ensemble model is used in the recognition system, including a type of feature representation, a base acoustic model, a base language model, one of their combinations, etc., and the best hypothesis is the one with the largest search score as defined by Eq. (5.8). Making decision at this level resembles the independent sample classification tasks in machine learning. This approach is straightforward to implement, as it requires no change in decoding engine, and the multiple decoding searches can potentially be parallelized. For long speech utterances, however, picking the one-best word sequence from a few alternatives is inefficient in reducing word errors. Zhang and Rudnicky [80] and Meyer and Schramm [57] investigated utterance level decision for boosted acoustic models, Shinozaki and Furui [67] also used the decision strategy in data sampling based ensemble acoustic and language models.

Another approach to utterance level combination is to let each system generate a set of n-best word sequence hypotheses, rescore the word sequence hypotheses by combining the scores of different models, and picking the hypothesis with the highest score. Ma et al. [53] took this approach when combining the models of multiple feature streams.

5.4.2.2 Word Level Combination

Word level combinations have been accomplished by several methods, including recognition output voting error reduction (ROVER), its extension to confusion-network combination (CNC), word pronunciation combination, as well as word prediction probability combination. The first two methods, ROVER and CNC, are applicable to combining recognition systems with arbitrary diversities, the third method is for combining acoustic models, and the last one is for combining language models.

5.4.2.2.1 ROVER

ROVER is a well-known method proposed by Fiscus [31]. In ROVER, the 1-best word sequence hypotheses of different systems for a speech utterance are aligned

through dynamic programming to generate a word transition network (WTN). A WTN is a linear graph, where between each pair of adjacent nodes is a set (or a bin) of aligned word arcs, with each arc corresponding to a word or an empty symbol in one sequence hypothesis, and the arcs are weighted by word confidence scores. For every such word set, a decision is made to choose the word hypothesis that maximizes an interpolated measure of word frequency and confidence score, like in Eqs. (5.2) and (5.3) when C represents the number of word candidates in the set. The time-ordered ROVER word hypotheses define the word sequence hypothesis for the speech utterance. A variant of ROVER is iROVER [40]. For each word set of a WTN, the word hypothesis is determined by an Adaboost classifier, which is trained from the WTNs' word sets and by using features extracted from word lattices of a development set. When the number of systems to be combined is small, iROVER works better than ROVER [40].

5.4.2.2.2 CNC

CNC is an extension to ROVER, where each recognition system's decoding search outcome is represented in a confusion network (CN) instead of a 1-best word sequence hypothesis, with the aim of presenting to the word-level combiner more alternative word hypotheses through the CNs [30]. The multiple confusion networks associated with the multiple systems are aligned by dynamic programming as in ROVER to generate a weighted linear graph (WLG), the weights being the word posterior probabilities in the CNs. Between each pair of adjacent nodes of the WLG is a set of words from which the best word hypothesis is to be determined. To do so, the posterior probabilities from the multiple systems are summed for each distinct word in a set, and the best word hypothesis is then determined as in Eq. (5.3). Again, the time-ordered word hypotheses define the word sequence hypothesis.

5.4.2.2.3 Word Pronunciation Combination

In the work of Meyer and Schramm [57], multiple acoustic models are generated, and for each acoustic model, a set of renamed phones is defined, each renamed phone being a variant of the corresponding phone in a standard phone set. For each vocabulary word, its pronunciation is augmented by the alternative pronunciations defined by the phone variants. In this way, each word has at least as many pronunciation variants as the number of acoustic models. During decoding search, a combined word score is computed by a weighted sum of the word scores based on its pronunciation variants of different acoustic models. To curb the increased search complexity due to the word score combination, a time-synchronous sum approximation is made. One-pass decoding search is realized with this method, whereas both ROVER and CNC require two-pass decoding search.

5.4.2.2.4 Word Prediction Probability Combination

When multiple language models are estimated by ensemble learning, the prediction probability for a word given a history can be combined from the probabilities for the word given the equivalent histories in the base language models. Xu and Jelinek [78] combined word n-gram probabilities in this way in their random-forest language model.

5.4.2.3 Subword Level Combination

In the work of Dupont and Bourlard [29], subword units of different time scales, i.e., phonemes and syllables, are modeled by MLP–HMMs, and during decoding search, the phoneme and syllable HMMs are kept as separate streams but their scores are forced to recombine at the syllable level. The log likelihood scores of the two streams are recombined either linearly or with a MLP. In the work of Dimitrakakis and Bengio [27], separate streams of phone segment scores computed from multiple acoustic models are forced to recombine at the phone level in a similar way as in Dupont and Bourlard [29], but the combination is based on the max rule, i.e., picking the largest score of one stream. The subword-level model combinations are suitable for one-pass decoding search.

5.4.2.4 State Level Combination

Multiple acoustic models $\Lambda^{(1)}, \cdots, \Lambda^{(K)}$ can be combined at the state level to give a combined likelihood score for each speech frame or feature vector o_t at each tied state s. This approach is also referred to as frame-level combination since the frame scores of the same state are combined rather than the segment scores of the same state as in word and subword level combinations. Within the realm of producing frame scores by using multiple models, a different approach is to stack up multiple simple MLPs vertically to improve the frame acoustic scores through deep learning, referred to as DSN. In the following, different methods for combining the frame scores of multiple models are first discussed, followed by a discussion on the approach of DSN.

5.4.2.4.1 Domain of Score Combination

Two methods are commonly used in frame score combination: weighted sum and weighted product. In the weighted sum, the likelihood scores of the individual models at each state s are linearly combined:

$$p\left(o_t\middle|s,\Lambda_s\right)=\sum_{i=1}^{K}\alpha_i\,p\left(o_t\middle|s,\Lambda_s^{(i)}\right) \tag{5.10}$$

where the combining weights $\alpha_i \geq 0$ and $\sum_{i=1}^{K}\alpha_i = 1$. When $p(o_t|s,\Lambda_s^{(i)})$'s are likelihoods of GMMs, the combined model for each state becomes an enlarged GMM. The weighted product rule is commonly used for combining scores of multi-stream features. For example, with K feature streams $o_t=\{o_t^{(1)},\cdots,o_t^{(K)}\}$, the likelihood scores of the individual streams are multiplicatively combined as

$$p\left(o_t\middle|s,\Lambda_s\right)=\prod_{i=1}^{K}p\left(o_t^{(i)}\middle|s,\Lambda_s^{(i)}\right)^{\alpha_i} \tag{5.11}$$

which is equivalent to linearly combining the log likelihood scores of the individual streams

$$\log p\left(o_t\middle|s,\Lambda_s\right)=\sum_{i=1}^{K}\alpha_i\log p\left(o_t^{(i)}\middle|s,\Lambda_s^{(i)}\right) \tag{5.12}$$

The product rule has also been used for just one stream of features, where the multiple scores come from multiple acoustic models instead. Since averaging the frame scores is done independently for each frame which does not need modification on the decoding search algorithm, the state-level combination is easier to implement than the word and subword level combinations in one-pass decoding search.

Instead of combining likelihood scores, combining posterior probability scores is often used for the MLP–HMM hybrid. To do so, the likelihood scores of the individual and the ensemble models in Eqs. (5.10) and (5.12) are replaced by their posterior probability counterparts of $p(s|o_t,\Lambda_s^{(i)})$ and $p(s|o_t,\Lambda_s)$, respectively. Averaging the log posterior probabilities has an appealing interpretation of minimizing the average Kullback–Leibler divergence between the ensemble posterior probability distribution and the individual model's posterior probability distributions [62].

Robinson et al. [62] investigated combining the posterior probability distributions in both linear and log domains on recurrent-neural-network (RNN) based RNN–HMM hybrid resulting from four combinations of two types of features and two recognition directions (forward, backward), where the log domain combination was shown to yield a better performance than the linear domain combination. Cook and Robinson [13] and Schwenk [66] investigated linearly combining frame phone posterior probabilities of multiple MLPs in boosted MLP–HMM acoustic models. Kingsbury and Morgan [46], Halberstadt and Glass [38], Wu et al. [77], McMahon et al. [56], Kirchhoff et al. [47], and Ma et al. [53] investigated combining state scores of multi-stream features from different perspectives, such as phone-scale feature vs. syllable scale feature, acoustic feature vs. articulatory feature, features of

different time-frequency spectral resolutions. Their results favored using the product combining rule for multi-stream feature based acoustic models. Dimitrakakis and Bengio [26] investigated both weighted sum and weighted product combining rule for simple phone-HMM multiple acoustic models. Xue and Zhao [79] investigated the weighted sum combining rule for ensemble acoustic model of GMM–HMM based on random forest of PDTs. More efforts of recent years on state-level score combination are mentioned in the subsequent discussions.

5.4.2.4.2 Combining Weights Estimation

The combining weights as given in Eqs. (5.10–5.12) are specific to the base models i, but they can also be specific to the state s or/and the feature vector o_t. The weights may be estimated as an integral part of ensemble model construction such as in boosting, and they may also be post estimated after base model construction or adaptively estimated from online data. As in classifier combination, the weights can also be specified according to certain rules. Setting the weights to be uniform, i.e., $\alpha_i = 1/K$, is simple and often gives reasonable results for a wide variety of ensemble learners.

McMahon et al. [56] used the minimum classification error criterion to estimate the weights for the multi-stream product based score combinations of Eq. (5.11). Xue and Zhao [79] investigated methods for determining the weights of Eq. (5.10). By viewing the α_i's as the weight parameters of a mixture density, MLE was performed on α_i's while holding the base models fixed. Moreover, they made the weights depend on o_t to emphasize the base models that were more discriminative for the current o_t. This was done by setting the weight of a base model to be proportional to the relative entropy (RE) or KL divergence between the tied-state posterior probability distribution of the base model for o_t and a uniform distribution. They also looked into using the n-best rule by assigning uniform weights $1/n$ to the base models that gave the n-best likelihood scores, which became the max rule when $n = 1$. For a conversational speech recognition task and with the random-forest ensemble acoustic model, they found the uniform weights to be consistently better than the n-best weights, and the MLE weights to be consistently better than the uniform and the RE weights. Combining MLE and RE weights led to further improvements for large ensemble models.

5.4.2.4.3 Deep Stacking Network

Deep stacking network (DSN) builds a deep structure in neural networks (NNs) by stacking up blocks of simple single-hidden-layer NNs [19, 20]. The bottom-level NN takes the raw input data as its input, while each higher-level NN takes the raw input data as well as the outputs of the lower blocks as its inputs, and the top NN

has an additional softmax layer that provides the phone-state posterior probabilities for each speech frame. A tensor-DSN (T-DSN) extends DSN by first generating two hidden representations through two linear transformations on the input and then combining the two hidden representations bilinearly to produce outputs [44]. Another extension to DSN is to use a kernel function in the input-to-hidden mapping (K-DSN) [21], which enables complete convex parameter learning and makes the effective number of hidden nodes in each hidden layer approach infinity.

5.4.2.5 Feature Level Combination

One approach to combining features is to construct a long feature vector for each frame by concatenating different types of feature vectors of the frame. Kirchhoff et al. [47] investigated concatenating MFCC and pseudo articulatory features followed by discriminative feature component selection to reduce the dimension of the combined feature vector. Zhu et al. [85] concatenated MLP features with PLP or MFCC features. Povey et al. [59] proposed an fMPE method that discriminatively projected posteriors of a large number of Gaussian models for a speech frame to a regular-sized feature space, and the projected features were then combined with PLP features additively to represent the frame.

In the case of MLP-based features, the posterior probability outputs from multiple MLPs for each o_t can be combined as the MLP feature of o_t. Chen and Zhao [11] trained multiple MLPs by cross-validation (CV) based data sampling (cf. Sect. 5.5.2.3) and used the averaged posterior probability features of the multiple MLPs to concatenate with MFCC features. Qian and Liu [60] used CV data sampling as well as different types of spectral features of o_t to construct multiple MLPs, and they generated the ensemble MLP feature by using another MLP to combine the posterior probability outputs of the base MLPs.

The above discussed methods for diversity exploitation are summarized in Fig. 5.2.

5.5 Ensemble Learning Techniques for Acoustic Modeling

A significant amount of diversity-generating efforts has been gravitated towards acoustic modeling, as it plays a central role in the accuracy performance of speech recognition. In this section, further discussions are given to those methods that have close relations with ensemble machine learning. These methods are categorized below as being explicit or implicit in generating diversity. Along the line of implicit diversity generation, the generic multiple system approach is also discussed, as it commonly involves differences in acoustic models among other possible difference factors.

Level of combination	Methods
Utterance	▪ Pick the word sequence with the highest score from multiple 1-best hypotheses ▪ Generate n-best word sequence hypotheses from each system, rescore the collection of multiple n-best hypotheses by combining scores of different systems, and pick the hypothesis with the highest score.
Word	▪ ROVER, iROVER ▪ CNC ▪ Combine word scores from different acoustic models by creating multiple pronunciations ▪ Combine word scores from different language models
Subword	▪ Recombine phone-stream and syllable stream scores at the syllable level ▪ Recombine phone streams of base acoustic models at the phone level
State	▪ Combine frame likelihood scores of each state from multiple feature streams and/or multiple acoustic models ▪ Combine frame posterior probabilities of each state from multiple feature streams and/or multiple acoustic models ▪ DSN, T-DSN
Feature	▪ Concatenate different types of features ▪ fMPE ▪ Average posterior probability features from multiple MLPs

Fig. 5.2 Combining multiple features, models, or systems in different levels of speech recognition

5.5.1 Explicit Diversity Generation

5.5.1.1 Boosting

Boosting algorithms as used in acoustic modeling are mostly the multiple class versions of Adaboost, i.e., Adaboost.M1 and Adaboost.M2 [35], since speech recognition mostly deals with multiple sound classes. In Adaboost.M1, the goal of the weak learner h^k, k being the learning iteration, is to minimize the training error for the resampled data distribution D^k, and h^k only generates a class label for each sample o_t. In Adaboost.M2, the goal of the weak learner h^k is to minimize a pseudo-loss measure with respect to a distribution over the resampled examples and their incorrect labels, and h^k outputs a vector of plausibility values pertaining to assigning o_t to the different classes. In theory, Adaboost.M1 requires each weak learner to have an error rate less than ½, while Adaboost.M2 requires each weak hypothesis to have its pseudo-loss slightly better than random guess.

One way to categorize the boosting methods that are adapted to acoustic modeling is based on their unit of resampling, where the common choices are either utterances or frames. Resampling utterances is straightforward to implement,

but it is not flexible for looking into the erroneous parts of an utterance. Because speech frames are not independent, resampling at the frame level requires either a simplifying assumption of independency or some additional formulations.

It is worth mentioning that boosting has been shown to be equivalent to functional gradient boosting [39]. Along this line, a boosted acoustic model can also be obtained through boosting the tied-state GMMs by adding in one Gaussian pdf at a time to a GMM while maximizing an explicit objective function. Several such efforts have been reported [28, 43, 71]. The details of these methods and results are omitted here for the sake of space.

5.5.1.1.1 Utterance Resampling

Assume a training set $U^0 = \{(O_i, h_i), i = 1, \cdots, N\}$, where O_i is the feature sequence of the i-th training utterance and h_i is its reference word sequence. Zhang and Rudnicky [80, 81] defined a pseudo-loss for Adaboost.M2 based on the n-best utterance hypotheses H of a speech recognizer:

$$\varepsilon^k = \frac{1}{2\left|H\right|\left|U^{k-1}\right|} \sum_{O_i \in U^{k-1}} \sum_{h \neq h_i} \left(1 - P_{\lambda_k}\left(h_i | O_i\right) + P_{\lambda_k}\left(h | O_i\right)\right)$$

where k index the boosting iteration, λ_k represents the acoustic model trained from the resampled training set U^{k-1}, h is one of the hypotheses in H generated by using λ_k for O_i, and the posterior probability $P_{\lambda_k}\left(h\middle|O_i\right)$ is approximated by

$$P_{\lambda_k}\left(h\middle|O_i\right) \approx \frac{P_{\lambda_k}\left(h, O_i\right)^{\beta}}{\sum_{h \in H} P_{\lambda_k}\left(h, O_i\right)^{\beta}}$$

with β an empirically set smoothing parameter. By minimizing the pseudo-loss ε^k, the model combining parameter c_k and the sample weights d_i^k are computed, and the resampled training set U^k obtained. The combining rule at the utterance level is

$$h^* = \arg\max_h \sum_{k=1}^{K} \left(\log \frac{1}{c_k}\right) P_{\lambda_k}\left(h\middle|O\right) \tag{5.13}$$

where $P_{\lambda_k}\left(h\middle|O\right)$ is the 1-best hypothesis from the model λ_k, or it is the re-ranked 1-best hypothesis, with re-ranking performed on the n-best list of λ_k by using features derived from language model and posterior probability scores of utterance, words, and phones. ROVER is also applied to the re-ranked 1-best hypotheses for word level combination.

On the task of CMU Communicator system, the authors showed that 3 or 4 base models were sufficient, beyond which no significant improvements were observed.

Using 4 models, utterance level combination reduced word error rate (WER) over the single model from 14.99 to 13.27 %, re-ranking n-best hypotheses followed by utterance level combination further reduced WER to 12.98 %, and applying ROVER on the re-ranked 1-best hypotheses led to the lowest WER of 12.52 %.

Meyer and Schramm [57] applied the resampling weights d_i^k of Adaboost.M2 directly to the acoustic model training criterion instead of generating resampled training utterance sets. The boosting weights modify the maximum likelihood (ML) training objective function as

$$F_{ML}^k(\lambda) = \sum_{i=1}^{N} d_i^k \log p\Big(O_i \,|h_i; \lambda\Big) \tag{5.14}$$

The boosted models are combined at the word level by using the multiple pronunciation approach discussed in Sect. 5.4.2.2.3, with the combining weights defined by boosting. Based on experimental results from a spontaneous medical report dictation task and the Switchboard task, the authors concluded that only 2 or 3 base models were needed, and boosting significantly reduced word error rate over a conventional single acoustic model system.

Schwenk [66] applied Adaboost.M2 to the MLP part of the MLP–HMM hybrid, and combined the frame phone posterior probability outputs of the base MLPs using the weights of boosting. On the OGI Numbers 95 continuous speech recognition task, boosting reduced word error rate from 6.3 to 5.3 %, which was significant for the task. Cook and Robinson [13] investigated heuristic procedures to boost the MLPs of an MLP–HMM hybrid and combined phone posterior probability outputs of the base MLPs by a weighted average. In one of the heuristic procedure, a subset of training utterances was first sampled from the training set to train a base MLP, which was used to compute the frame error rate on the unused training utterances, and the utterances with high error rates were next used to train a second MLP. The authors experimented on the Wall Street Journal tasks of Hub2 93 and Hub3 95 and achieved sizable word error reductions by combining two such base MLPs.

5.5.1.1.2 Frame Resampling

Zweig and Padmanabhan [86] used Adaboost.M2 and frame level resampling to boost the GMMs of context-dependent phones (CDP). Assume a total of C CDPs and denote the GMM of the yth CDP by $f_y(o)$. The classifier of the kth iteration computes the CDP posterior probabilities given a frame sample o as

$$h_k(o, y) = P^k\left(y\middle|o\right) = \frac{P(y) f_y^k(o)}{\sum_{c=1}^{C} P(c) f_c^k(o)}, \quad y = 1, \ldots, C.$$

and the $h_k(o, y)'s$ are combined as in Eq. (5.13) for frame classification. As C is usually very large, the authors considered restricting the CDPs to a small subset for a given o, or boosting clusters of CDPs. On a voicemail transcription task, this boosting strategy gave a small and yet consistent improvement to frame classification accuracy as well as a reduction in word error rate.

Zhang and Rudnicky [81] also used Adaboot.M2 for frame level resampling. A frame-level word classifier is approximated from the n-best utterance hypotheses of each base model to make the frame weights relevant to word errors. In each boosting iteration, a resampled training set is generated to train a base acoustic model, and the base models are combined at the utterance level as discussed in Sect. 5.5.1.1.1. The authors showed that this method reduced word errors but it was not as effective as the utterance level resampling discussed in Sect. 5.5.1.1.1.

Saon and Soltau [64] modified Adaboost.M1 to boost acoustic models. The feature vectors o_t are classified by a full acoustic model and a unigram language model, unlike the independent frame classification of Zweig and Padmanabhan [86]. The resampling weights are applied to the state occupancy counts in GMM parameter estimation, as well as in phonetic decision tree construction. The base acoustic models are combined at the word level by ROVER and at the frame level by using a weighted sum of log likelihood scores. On English and Arabic broadcast news tasks, the authors showed significant word accuracy gains of 1 % absolute for both the maximum likelihood and discriminatively trained base acoustic models with 50 h of training speech data; but the benefit of boosting was reduced when a much larger amount of data of 2,000 h was used to train the acoustic models.

Dimitrakakis and Bengio [27] proposed an expectation boosting method and performed resampling at both utterance and frame levels. An ad hoc loss function relating to word error rate in each utterance is defined. Utterances are resampled to form new training sets as in Cook and Robinson [13]. The frames associated with a word error in an utterance are marked, and the weights of the marked frames are computed based on an error distribution model, and the frame weights are applied to the state occupancy counts in HMM parameter estimation. The base models are combined at the state level using the weighted sum method. The authors compared this approach with their earlier work of using Adaboost.M1 for pre-segmented phone segments [26] on the OGI Numbers task. They found that the expected boosting gave better performance than the phone segment boosting, but the improvements over an ML single acoustic model system were only moderate.

5.5.1.2 Minimum Bayes Risk Leveraging (MBRL)

Bresline and Gales [6] proposed using Minimum Bayes Risk training with a modified loss function to generate complementary acoustic models. The loss function reflects how well each training speech word is modeled by the base models that are generated so far, and the loss values are the posterior probabilities of the erroneously

hypothesized words given by the combined existing base models. To compute the word posterior probabilities in a speech utterance, a confusion network is generated by each base model, and the word posterior probabilities are accumulated as in CNC. The loss values are further quantized to 0–1, where the erroneous words' posterior probabilities larger than a threshold are set to 1, and the rest are given a loss value of 0. This importance-based data sampling for sequentially training the base models resembles boosting in spirit but is designed directly for speech model training. The authors evaluated MBRL on a Broadcast News Mandarin speech recognition task by combining the models at the word level with CNC. They found that MBRL slightly improved a ML single model baseline, while using different feature transformation frontends for the base models of MBRL increased model diversity and word error reduction.

5.5.1.3 Directed Decision Trees

Bresline and Gales [7] also proposed a sequential method of directed decision trees (DDT) to bias the phonetic decision tree construction towards separating the states of confusable words. Toward this end, a word loss is defined as one minus the word posterior probability, sharpened or dampened by a power parameter, and the loss is applied to the state occupancy counts in model training to help a new model discriminate the word from its confusable ones. The word posterior probabilities are obtained from CNC, and model combination is again based on CNC. On a Broadcast News Arabic speech recognition task, by using three models, including a baseline model, the DDT approach achieved 1–1.2 % absolute word error reductions on test data that mismatched training conditions.

5.5.1.4 Deep Stacking Network

Although driven by different perspectives and realized in different forms, DSN shares a commonality with boosting in terms of introducing base classification modules sequentially to build an accurate classifier. On the training set, DSN possesses the property that each higher NN block is guaranteed to perform better than its lower blocks due to its retaining the raw input for each higher NN block [44]. Because batch learning can be used for estimating the input-to-hidden and hidden-to-output mapping parameters, a parallel implementation is feasible, making DSN scalable to large amount of training data. T-DSN gave lower phone error rate on TIMIT than DNN did. DSN and K-DSN has also shown promising potentials in application tasks of spoken language understanding, dialog state tracking, and information retrieval [21–23, 73].

5.5.2 *Implicit Diversity Generation*

5.5.2.1 Multiple Systems and Multiple Models

In ROVER, Fiscus [31] combined 1-best utterance hypotheses from five systems developed at BBN, CMU, CU-HTK, Dragon, and SRI for NIST's LVCSR 1997 Hub 5-E Benchmark Test Evaluation. Since the research sites all have long histories in speech recognition, the five systems naturally had different traits in features, models, and decoding search. An illuminating point made in Fiscus' paper was that even though the word error rates of two systems differed only by 0.2 % (44.9 vs. 45.1 %), out of 5,919 errorful segments from the two systems, 738 segment errors were unique to one system and 755 segment errors were unique to another system, or about 25% errorful segments had no overlaps between the two systems. This difference in error patterns supports the notion that different systems often implicitly carry diversity, and combining them to exploit the diversity can help reduce word errors. Indeed, ROVER achieved impressive word error reductions by combining the five systems. While the original five systems had individual word error rates of 44.9, 45.1, 48.7, 48.9, and 50.2 %, respectively, which averaged to 47.56 %, the word error rates from ROVER were 39.7, 39.5, and 39.4 % when using vote by frequency of occurrence (FOO), vote by FOO and average word confidence, and vote by FOO and maximum confidence, respectively.

Evermann and Woodland [30] generated multiple systems by varying the acoustic model training criterion between MLE and MMIE as well as the context width of triphone and quinphone, and they used CNC to combine the four systems resulting from the combinations of two training criteria and two context widths. On the task of NIST Conversational Telephone Speech evaluation (eval00), the four systems had the Viterbi 1-best word error rate of 28.8, 28.4, 27.6, and 27.3 %, respectively, and the CN 1-best word error rate of 27.8, 27.2, 26.9, and 26.5 %, in that order. By combining the four systems, CNC reduced the word error rate to 25.4 %. The authors also showed that ROVER obtained lower word errors by combining the CN 1-best hypotheses instead of the Viterbi 1-best hypotheses.

ROVER and CNC based system combination have since attracted many efforts which cannot be fully enumerated here. But the study of Gales et al. [37] is discussed here to compare the effects of generating multiple systems by varying the acoustic models versus actually using multiple systems of different research sites. The authors investigated combining multiple systems using ROVER and CNC for a Broadcast News transcription task in the DARPA EARS program, and they compared multiple systems of two types. The first type was based on varying acoustic models, including speaker adaptive model, gender-dependent model, etc., as well as on varying speaker segmentations of audio recordings from LIMSI, BBN, and CUs. They found that different segmentations had a larger effect than different models on word error rate. The second type was to combine four cross-site systems of BBN, LIMSI, SRI, and CU, which made a much larger impact on word error rate than the first type. On the Eval04 task, the four systems had the individual word error

rates of 12.8, 14.0, 14.6, and 12.8 %, while their combination reduced the word error rate to 11.6 %. This outcome supports the notion of incorporating many difference factors in a system ensemble, since the interactions among different factors, such as speech segmentation, feature analysis, acoustic model, and language model etc. may potentially produce larger differences in word error patterns across systems than varying only one or two factors.

It is worth mentioning here yet another system combination scenario considered by Wachter et al. [75] that differs from most of the studied cases. The authors combined a statistical acoustic model based speech recognition system with a speech template based recognition system by using the rescoring approach for utterance level combination. Although the template-based system had a larger word error rate than that of the statistical acoustic model based system, because the error patterns of the two types of systems were different, their combination reduced word errors below the statistical model based system.

5.5.2.2 Random Forest

Siohan et al. [68] first proposed randomizing phonetic decision trees to generate multiple acoustic models, and they used ROVER to combine the 1-best hypotheses of the multiple systems based on the different acoustic models. Like the method of [24], the PDTs are randomized by randomly selecting a node split out of n-best possible splits for each node. The degree of randomness is controlled by the parameter n: using a large n would infuse large randomness into a tree, but the tree quality might be compromised by some bad node splits, and the opposite is true when n is small. The conventional PDT is resulted when $n = 1$, i.e., always select the best split for each node. Evaluations were carried out on MALACH and several test sets of the DARPA EARs project, where consistent and significant word error reductions were obtained when two or more systems using the randomized acoustic models were ROVERed with a baseline system. Although increasing n led to poorer individual systems, the accompanied rise in randomness increased diversity of the acoustic models and thus helped reduce word errors.

Along the direction of randomizing the phonetic decision trees, Xue and Zhao [79] proposed randomly sampling phonetic questions from a full set of questions to generate multiple question subsets and using each sampled question subset to construct the PDTs of one base acoustic model as in conventional model training. They combined the multiple acoustic models at the state level by using the weighted sum of Eq. (5.10) for the random-forest defined tied triphone states. Specifically, given a question set Ω with M questions and a parameter $M' < M$, a question subset is generated by random sampling Ω without replacement M' times, and repeating this procedure K times generates K subsets of questions. The parameter M' controls the degree of randomness: using a large M' would make the question subsets to differ less from one another, but it is good for the quality of the individual PDTs and thus the base acoustic models, and vice versa. This approach of constructing random forests of PDTs bears a resemblance with the random subspace approach

of Ho [41], but it differs from that of Ho in that each random subset of questions is used to train one acoustic model which itself has a large number of PDTs rather than a single tree, and the random forest defines multiple clustering structures of triphones rather than combined classification votes or regression predictions.

The authors evaluated the RF-based ensemble acoustic model on a conversational speech recognition task of telehealth captioning [82]. Compared with an ML single acoustic model baseline system having a word accuracy of 78.96 %, the ensemble acoustic model with 10 base models improved word accuracy to 81.93 %, giving an absolute accuracy gain of 2.97 %, or a relative word error reduction of 14.12 %. As discussed below in Sects. 5.7 and 5.8, using the RF ensemble acoustic model with state-level weighted sum combination offers benefits in triphone tying and mixture modeling, and it is also convenient for redundancy removal.

5.5.2.3 Data Sampling

Shinozaki and Furui [67] investigated using utterance clustering to partition training data into disjoint subsets and from which to train multiple acoustic models and multiple language models. These models are used in a large number of parallel decoding systems, one for each combination of the acoustic and the language models, and the systems are combined at the utterance level by picking the word sequence hypothesis of the largest decoding score as discussed in Sect. 5.4.2.1. Utterance clustering is based on a K-means like procedure, with an effort on keeping the cluster sizes balanced. Speech data are initially randomly partitioned to K clusters, and K cluster-specific acoustic models are produced by adapting a general acoustic model to the data in each cluster. Cluster assignment criterion is maximum likelihood using the adapted acoustic models, and the models are readapted for each new clustering. Text clustering on lecture speech transcript data proceeds in a similar way with the cluster assignment criterion being minimum bigram perplexity, and that cluster-specific language models are estimated by mixing the cluster specific data with the full set of training data. The authors reported results on a Japanese lecture speech transcription task, where the combination of ten acoustic models and ten language models amounting to 100 parallel systems reduced word error rate from 24.9 to 22.0 %.

Chen and Zhao [10–12] investigated utterance-level data sampling methods for training multiple acoustic models, including cross-validation (CV) data partition, overlapped speaker clustering, and random sampling without replacement. This data sampling approach also produces a random forest of phonetic decision trees for each phone state, with the models combined in the same way as in [79]. In the CV method, K-fold cross-validation based data partition is used to generate K overlapped training subsets, with each having a $(K-1)/K$ fraction of total training utterances, and from each subset an acoustic model is trained. More than K subsets can also be generated by a K-fold CV via shifting the data partition points while keeping the subset fraction as $(K-1)/K$. In speaker clustering, K-medoids clustering with the maximum likelihood cluster assignment criterion is used.

The clustering approach differs from that of Shinozaki and Furui [67] in that the clusters are overlapped and the amount of overlap is controlled by the desired subset size for training the base models, the utterances of each speaker are always assigned to the same cluster, and the acoustic models are directly trained instead of adapted for the individual clusters.

On TIMIT phone recognition, fixing the ensemble size K, CV and speaker clustering both performed better than random sampling, and in comparison with CV, speaker clustering allowed a smaller amount of data overlap while generating the same level of word accuracy (with $K = 10$ the contrast was 50 % overlap for clustering vs. 90 % overlap for CV). An encouraging result was that the positive effect of ensemble model was amplified or maintained as the speech features and model trainings improved for the base models. Enlarging the acoustic model ensemble by varying the GMM mixture sizes in different base models also produced positive effects. On the telehealth captioning task [82], by combining a tenfold CV ensemble model using GMMs of size 16 with another tenfold CV ensemble model using GMMs of size 32, word accuracy was improved by 3.3 % absolute from 79.2 % of a maximum likelihood single model baseline system.

Cui et al. [14] investigated a feature bootstrapping scheme for acoustic modeling. The PDTs that are trained from MFCC based features are fixed as in conventional single acoustic modeling, but for each PDT defined tied state, feature-specific GMMs are estimated separately for different types of features, including MFCC, PLP, and LPCC. The multiple feature-specific GMMs are combined at the state-level using weighted log likelihood scores of Eq. (5.12). The authors reported results on the DARPA Transtac project for speech-to-speech translation and showed superior performance over the single models of the three feature types.

Dimitrakakis and Bengio [27] used bootstrap sampling on speech utterances to generate multiple phone HMMs, and combined the multiple model frame scores by a weighted sum of Eq. (5.10) or a weighted product of Eq. (5.12) for each HMM state. On the OGI Numbers 95 task, they showed that the weighted sum combination reduced word error rate over using single phone HMMs. They also compared the effect of boosting and bagging on the same task, and found that while boosting worked well for phone segment classification, it was not as effective as bagging on phone recognition.

5.6 Ensemble Learning Techniques for Language Modeling

Xu and Jelinek [78] proposed a random forest based n-gram language modeling approach to smooth word prediction probabilities. In a decision tree based n-gram language model, a word prediction tree is constructed to classify the (n–1)-word histories into some equivalent histories, which greatly reduces the number of word prediction parameters and thus alleviates the problem of data sparseness. In the random-forest language model, the decision trees are randomized through randomly sampling training sentences, randomly selecting node-split questions, as

well as randomly initializing data partition at each node, using entropy as the node split goodness measure. The random forest of decision trees defines a multi-way clustering of the (n–1)-word histories, and the word prediction probabilities from the different trees are linearly combined. The authors applied the RF language models with 10–100 decision trees to the speech recognition tasks of Wall Street Journal and IBM 2004 conversational telephony system for rich transcription, and they obtained 0.6–1.1 % absolute word error rate reduction over the commonly used modified Kneser-Ney smoothing.

Recurrent neural network (RNN) based language modeling has gained momentum in recent years. Mikolov et al. [58] investigated training multiple RNN language models (RNN–LM) and linearly combining their word prediction probabilities. The multiple RNNs are generated by varying the RNN initialization weights, the size of hidden layers, as well as the learning rate in adaptive RNNs, and the RNNs are also combined with the n-gram language models based on the modified Kneser-Ney back-off and the random forest as discussed above. The authors showed that on the Penn Treebank portion of the Wall Street Journal task, the combined models worked better than any individual RNN–LMs, and they also showed large improvements over the Kneser-Ney back-off based n-gram language model in perplexity and word error rate.

As discussed in Sect. 5.5.2.3, Shinozaki and Furui [67] investigated using sentence clustering to generate multiple trigram language models together with using utterance clustering to generate multiple acoustic models. A noteworthy point shown through their work is that the ensemble learning gains from language modeling and from acoustic modeling are largely complementary. Specifically, in their lecture transcription task, the baseline word error rate of the single acoustic model and single language model based system was 24.9 %; combining ten acoustic models but still using one language model reduced word error rate to 23.0 %; combining ten language models but still using one acoustic model reduced word error rate to 23.6 %, and combining the ten acoustic models and the ten language models reduced word error rate to 22.0 %.

5.7 Performance Enhancing Mechanism of Ensemble Learning

5.7.1 Classification Margin

Schwenk [66] and Meyer and Schramm [57] examined the effect of boosting acoustic models on classification margins. Schwenk showed that while there was the trend that classification margins improved with the boosting iterations, the margins of some samples remained totally wrong throughout the boosting iterations, indicating the difficulties that the outlier samples might cause on boosting. Meyer and Schramm showed that boosting improved margins over their baseline ML

model, and boosting discriminatively trained models did even better on the margins. Chen and Zhao [10] examined the effect of data sampling based ensemble acoustic model on classification margin and showed that a tenfold CV based ensemble acoustic model largely improved classification margin over an ML single acoustic model.

5.7.2 Diversity

As the potential performance of an ensemble model is implied by the differences among its base acoustic models, several authors defined diversity measures for acoustic models and evaluated this attribute in conjunction with word error rates.

In Xue and Zhao [79], the correlation between two acoustic models is defined to be the average correlations between their corresponding triphone state posterior probabilities, and the ensemble correlation to be the average pairwise base model correlations. In their random forest acoustic modeling work discussed in Sect. 5.5.2.2, the authors showed that the ensemble correlation reduced with the decrease in the question subset size M', and good word accuracy performances occurred when M' was in the range of 70–90 % the total number of questions M, which amounted to 0.82–0.89 ensemble correlations (measured on one speaker's test speech). Generally speaking, a good balance between the base model quality and the ensemble diversity may be attained by properly choosing M'. When M' is too small, the reduced base model quality would negatively offset the positive effect of the increased ensemble diversity, and when M' is too large, the increased ensemble correlation or reduced diversity would negatively offset the positive effect of improved base model quality.

In Chen and Zhao [12], three measures for evaluating diversity in an ensemble acoustic model are defined, including standard deviation of frame likelihood scores, classification agreement on phone segments, and KL distance of triphone HMM states. In frame-score standard deviation, speech frame likelihood scores are first computed by base models with respect to a common state sequence which is defined by Viterbi alignment from a baseline model, and per frame standard deviation of the frame scores are computed and then averaged over all frames. In phone classification agreement, if two base models produce an identical phone label for a phone segment, which is again defined by a common Viterbi alignment, then the agreement count is incremented by 1, and the classification agreement is the count accumulated over all phone segments and model pairs, normalized by the total numbers of phone segments and model pairs. In phone-state KL distance, the Kullback–Leibler divergence between two GMMs in the same triphone HMM state of each base model pair is computed, and the KL distances are averaged over the states and triphones as well as the model pairs to represent the ensemble diversity. On the TIMIT task, the authors compared the diversity measures alongside the phone accuracies of the base and the ensemble models for five scenarios that differed in speech features (MFCC vs. MFCC + MLP posterior probabilities), model training criteria (MLE

vs. MPE), data sampling methods (tenfold CV vs. speaker clustering), as well as certain combinations of these factors. The study confirmed that the performance of an ensemble model with a low base model quality and a high diversity could surpass an ensemble model with a high base model quality and a low diversity, and that discriminative feature of MLP, discriminative model training, and speaker clustering all helped increase diversity.

In Bresline and Gales [7], decision tree divergence is defined for the multiple acoustic models generated by the directed decision trees (DDT) as discussed in Sect. 5.5.1.3. The tree divergence uses Gaussian density divergence as a surrogate. For each pair of acoustic models, any triphone state falls under two leaf nodes of two DDTs whose leaf nodes are each modeled by a Gaussian density, and the tree divergence is obtained by first computing a symmetrized KL-divergence between two Gaussian pdfs for each triphone state and then averaging the divergence over all triphone states by using the state posterior probabilities as the weights. The authors showed large tree divergence values between the DDT acoustic models and the baseline acoustic model when the Bayes loss used in the tree construction was weighted by a power parameter greater or equal to one.

In Audhkhasi et al. [1], ambiguity decomposition of Krogh and Vedelsby [48] is used with a 1-of-C word encoding to decompose an approximate ROVER word error rate (WER) into a difference of two terms: the average WER of the individual systems minus the diversity among these systems, where the diversity term is defined similarly as the second term in Eq. (5.7), and C is the vocabulary size in a word set of ROVER's WTN. The approximate nature of the decomposition for ROVER is due to the fact that the combined regression function $H(x)$ of Eq. (5.7) is a weighted average of the base regression functions, while in ROVER a hard decision is made on the word hypothesis, which amounts to setting $H(w)$ to 1 for $w = \arg\max_{w'} \sum_{i=1}^{K} \alpha_i h_i(w')$ and resetting $H(w')$ to 0 for $w' \neq w$. The authors derived error bounds for the approximate decomposition with respect to the true ROVER WER and validated it on speech recognition tasks of BN HUB4, WSJ, and ICSI by ROVERing three recognition systems that used acoustic models trained with different criteria or transformation and keeping 10-best hypotheses from each system. The authors also proposed taking the diversity term as an objective to construct multiple acoustic models and showed its connection with the MBRL approach of Breslin and Gales [6].

5.7.3 *Bias and Variance*

In Xue and Zhao [79], the effect of the random forest acoustic model (Sect. 5.5.2.2) on triphone state tying and GMM resolution is discussed. In an ensemble of K acoustic models, since a triphone state is tied to different clusters in different base models and the combined model for the triphone state is consisted of K GMMs

from its K tied states, the triphone state is in effect modeled by a larger GMM that is jointly defined by K state-tying clusters. The combination of the state tying structures across base models defines many more uniquely combined GMMs than the original single model GMMs, and thus the granularity of state tying is effectively refined by the random forest, and the combined states underlying the unique GMMs are referred to as RF-tied states. The authors showed that on one speaker's data in the telehealth dataset [82], the number of tied states in the baseline single acoustic model was 1,603, while the number of RF-tied states was 12,033, indicating that the combined model could distinguish 7.5 times more tied triphone states than the baseline single model did. On the other hand, because the data in each RF-tied state came from the leaf nodes of K different PDTs, having the large number of GMMs did not suffer from data sparsity. When measured on the same speaker's dataset, in the baseline acoustic model, each tied triphone state cluster had on average 9.1 triphone states, and in the ensemble model each RF tied state had on average 29.8 triphone states. Both the increased state-tying resolution and the increased acoustic space coverage due to the larger number of Gaussian pdfs in each GMM indicate a bias reduction effect by the random forest acoustic model.

Another phenomenon illustrated in Xue and Zhao [79] is that overfitting the individual acoustic models does not lead to overfit in the ensemble model. The authors showed that for the baseline model, when varying the mixture size of the GMMs as 8, 16, 20, and 24, word accuracy was 77.65, 78.96, 78.68, and 78.15 %, respectively, where the fact that the performance peaked at mixture size 16 suggested an underfit by the smaller size and an overfit by the larger sizes. However, for the ensemble acoustic model with $K = 20$ base models, the word accuracies were 78.06, 80.81, 81.86, and 81.92 % respectively for the base model mixture sizes of 8, 16, 20, and 24, i.e., the ensemble performance increased with the mixture size. This implies that the ensemble model is able to take advantage of the reduced bias in the overfit base models by reducing the variance that accompanies the overfit.

5.8 Compacting Ensemble Models to Improve Efficiency

While an ensemble system improves recognition accuracy in general, it also requires more memory storage and computation power than a conventional single system does. For resource limited applications such as mobile computing, certain post processing can be applied on an ensemble system to reduce its redundancy. Along this line, several criteria have been proposed to cluster the component models of ensemble models, mostly for the weighted-sum combined GMMs, resulting in significant reduction in model size while largely maintaining the performance edge of ensemble learning.

5.8.1 Model Clustering

Xue and Zhao [79] examined the Gaussian pdfs of the combined GMMs for the RF tied states (cf. Sect. 5.7.3) in the PCA projected 2-D feature space, and they showed that the combined GMMs provided a denser coverage of the feature space but also increased overlaps and thus redundancies among the Gaussian pdfs. The authors investigated clustering the Gaussian pdfs to produce fewer but more representative ones for the GMMs of each RF-tied state. The K-means and agglomerative clustering methods were used for this purpose with a Bayes-error-based dissimilarity measure that characterized the overlapped area under two Gaussian curves. On the same task as discussed in Sects. 5.5.2.2 and 5.7.3, through the clustering the ensemble system's decoding search time approached that of the single model system while the ensemble word accuracy edge was largely retained. When agglomerative clustering was applied to an ensemble of 100 base models with the size-16 GMMs used in the base models, which amounted to 1,600 Gaussians in each combined GMM of a RF-tied state, keeping 32 Gaussians after the clustering (a compression ratio of 50) gave an absolute word accuracy gain of 1.56 % over the baseline and a decoding time that was 1.4 times the baseline time, and keeping 16 Gaussians after the clustering (a compression ratio of 100) gave an absolute word accuracy gain of 1.04 % over the baseline and a decoding time that was 1.2 times the baseline time.

5.8.2 Density Matching

Cui et al. [16] also investigated reducing redundancy in weighted-sum combined GMMs and they proposed two additional steps after clustering full-covariance Gaussian pdfs. The first additional step was to minimize the KL divergence between each clustered full-covariance GMM with its original combined GMM by re-estimating the parameters of the clustered GMM so as to closely approximate the original combined GMM. For this purpose, they used methods of variational EM and Monte Carlo. The second additional step was to convert the reestimated full-covariance Gaussian pdfs to diagonal-covariance Gaussian pdfs to reduce computations in decoding search, and a Monte Carlo method was again used to estimate the parameters of the diagonal-covariance Gaussian pdfs while using the full-covariance GMM as the reference, and the obtained parameters of the diagonal Gaussian pdfs were further optimized by performing a maximum likelihood estimation on resampled speech utterances. The authors showed on a low-resource Dari speech recognition task that when the ensemble model was restructured to have the same number of diagonal Gaussian pdfs as in a maximum likelihood single-model baseline, absolute word error reductions of 2.9 and 1.9 % were maintained over the baseline on a held-out test set and live evaluation data, respectively. As a comparison, using the full-covariance combined model that had nine times more

Gaussian pdfs than the baseline gave absolute WER reductions of 4 and 3.7 %, respectively, over the baseline. The authors also evaluated the restructured model against a single model baseline when the acoustic models were both discriminatively trained and they showed absolute WER reductions of 1.4 and 1.3 %, respectively, over the baseline.

5.9 Conclusion

Ensemble learning in speech recognition has been progressing in parallel with efforts in machine learning and it is currently an active direction of research. The complex nature of speech recognition gives ensemble learning many opportunities for exploration. In this chapter, a number of successful approaches that have emerged in this area and targeted at different levels and components of speech recognition are discussed. While some of the approaches have apparent counterparts in machine learning, they are seen to be also crafted to address the unique problems in speech recognition.

One important issue in ensemble learning is explicit versus implicit diversity generation. For speech recognition, both directions have been explored and shown successful in acoustic modeling. The two approaches appear to differ in the number of base models that are needed for an ensemble model. The explicit methods covered here invariably used just two or a few base models, while the implicit methods benefitted from a much large number of base models. This dichotomy may be attributed to the fact that the explicit approach derives diversity directly from the training error patterns and just a few base models are sufficient to cover the major error patterns, while the implicit approach relies more on exploiting certain randomness aspects in training data to cope with the variations in unseen data, and as such it may not be as well focused as the explicit approach but could be more flexible nevertheless.

Systematically generating diversity in a certain component of a speech recognition system, e.g., acoustic model, versus utilizing diversity of multiple factors indirectly across different speech recognition systems appear to be an interesting topic. Combining multiple difference factors across systems appear to be effective in generating complementary word error patterns while manipulating a single component of a system may not be always as successful. On the other hand, the two approaches have different implications in decoding search, where combining models can be realized in one-pass decoding search such as frame score combination, combining systems requires multiple-pass decoding search such as ROVER or CNC.

The score combining methods of weighted sum versus weighted product is also an issue worthy of attention. In the case of working with different feature streams, the weighted product combining rule has been favored over the weighted sum, while in the case of combining base models of GMM–HMMs and for the same features, the weighted sum combining method may have certain advantages. Further analyses and thorough experimental comparisons may be needed to draw firm conclusions.

In addition to the predominantly supervised learning work discussed herein, ensemble learning efforts are also being directed toward utilizing abundantly found data for semi-supervised acoustic model training [15]. Overall, ensemble learning for speech recognition has been successful. It is expected that by keeping pace with the advances in machine learning, speech and language modeling, as well as computing technology, continued efforts along this direction will further boost the accuracy and robustness of speech recognition.

References

1. K. Audhkhasi, A.M. Zavou, P.G. Georgiou, S.S. Narayanan, Theoretical analysis of diversity in an ensemble of automatic speech recognition systems. IEEE Trans. ASLP **22**(3), 711–726 (2014)
2. P. Biuhlmann, Bagging, subagging and bragging for improving some prediction algorithms, in *Recent Advances and Trends in Nonparametric Statistics*, ed. by E.G. Akritas, D.N. Politis (Elsevier, Amsterdam, 2003)
3. J.K. Bradley, R.E. Schapire, FileterBoost: regression and classification on large datasets, in *Advances in Neural Information Processing Systems*, ed. by J.C. Platt et al., vol. 20 (MIT Press, Cambridge, 2008)
4. L. Breiman, Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
5. L. Breiman, Random forests. Mach. Learn. **45**, 5–32 (2001)
6. C. Bresline, M.J.F. Gales, Generating complimentary systems for large vocabulary continuous speech recognition, in *Proceeding of Interspeech* (2006)
7. C. Bresline, M.J.F. Gales, Building multiple complementary systems using directed decision tree, in *Proceeding of Interspeech* (2007), pp. 1441–1444
8. G. Brown, An information theoretic perspective on multiple classifier system, in *Proceedings of MCS* (2009), pp. 344–353
9. S.F. Chen, J.T. Goodman, An empirical study of smoothing techniques for language modeling, in *Proceedings of ACI* (1996)
10. X. Chen, Y. Zhao, Data sampling based ensemble acoustic modeling, in *Proceedings of ICASSP* (2009), pp. 3805–3808
11. X. Chen, Y. Zhao, Integrating MLP feature and discriminative training in data sampling based ensemble acoustic modeling, in *Proceeding of Interspeech* (2010), pp. 1349–1352
12. X. Chen, Y. Zhao, Building acoustic model ensembles by data sampling with enhanced trainings and features. IEEE Trans. ASLP **21**(3), 498–507 (2013)
13. G. Cook, A. Robinson, Boosting the performance of connectionist large vocabulary speech recognition, in *Proceeding of ICSLP* (1996), pp. 1305–1308
14. X. Cui, J. Xue, B. Xiang, B. Zhou, A study of bootstrapping with multiple acoustic features for improved automatic speech recognition, in *Proceeding of Interspeech* (2009), pp. 240–243
15. X. Cui, J. Huang, J.-T. Chien, Multi-view and multi-objective semi-supervised learning for HMM-based automatic speech recognition. IEEE Trans. ASLP **20**(7), 1923–1935 (2012)
16. X. Cui, J. Xue, X. Chen, P. Olsen, P.L. Dognin, V.C. Upendra, J.R. Hershey, B. Zhou, Hidden Markov acoustic modeling with bootstrap and restructuring for low-resourced languages. IEEE Trans. ASLP **20**(8), 2252–2264 (2012)
17. G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. ASLP **20**(1), 30–42 (2012)
18. L. Deng, D. Sun, A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory feature. J. Acoust. Soc. Am **95**(5), 2702–2719 (1994)

19. L. Deng, D. Yu, Deep convex network: a scalable architecture for speech pattern classification, in *Proceeding of Interspeech* (2011)
20. L. Deng, D. Yu, J. Platt, Scalable stacking and learning for building deep architectures, in *Proceeding of ICASSP* (2012a)
21. L. Deng, G. Tur, X. He, D. Hakkani-Tur, Use of Kernel deep convex networks and end-to-end learning for spoken language understanding, in *IEEE workshop on spoken language technologies* (2012b)
22. L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, A. Acero, Recent advances in deep learning for speech research at Microsoft, in *Proceeding of ICASSP* (2013a)
23. L. Deng, X. He, J. Gao, Deep stacking networks for information retrieval, in *Proceeding of ICASSP* (2013b)
24. T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization. Mach. Learn. **1**(22), 139–157 (1998)
25. T.G. Dietterich, Ensemble methods in machine learning, in *Proceeding of MCS* (2000), pp. 1–15
26. C. Dimitrakakis, S. Bengio, Boosting HMMs with an application to speech recognition, in *Proceeding of ICASSP* (2004), pp. V-621–624
27. C. Dimitrakakis, S. Bengio, Phoneme and sentence-level ensembles for speech recognition. Eurasip J. ASMP (2011). doi:10.1155/2011/426792
28. J. Du, Y. Hu, H. Jiang, Boosted mixture learning of Gaussian mixture HMMs for speech recognition, in *Proceeding of Interspeech* (2010), pp. 2942–2945
29. S. Dupont, H. Bourlard, Using multiple time scales in a multi-stream speech recognition system, in *Proceeding of Eurospeech* (1997), pp. 3–6
30. G. Evermann, P.C. Woodland, Posterior probability decoding, confidence estimation and system combination, in *Proceeding of speech transcription workshop* (2000)
31. J.G. Fiscus, A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER), in *Proceeding of IEEE ASRU Workshop* (1997), pp. 347–352
32. A. Fred and A. K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. PAMI, **27**(6), 835–850 (2005)
33. J. Friedman, P. Hall, On bagging and nonlinear estimation. J. Stat. Plan. Inference **137**(3), 669–683 (2007)
34. J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting. Ann. Stat. **28**(2), 337–407 (2000)
35. Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in *Proceeding of ICML* (1996), pp. 1–9
36. Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
37. M. Gales, D. Y. Kim, P. C. Woodland, H. Y. Chan, D. Mrva, R. Sinha, and S. E. Tranter, Progress in the CU-HTK broadcast news transcription system, IEEE Trans. ASLP, **14**(5), 1513–1525, (2006)
38. A.K. Halberstadt, J.R. Glass, Heterogeneous measurements and multiple classifiers for speech recognition, in *Proceeding of ICSLP* (1998), pp. 995–998
39. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, 2001)
40. D. Hillard, B. Hoffmeister, M. Ostendorf, R. Schluter, H. Ney, iROVER: improving system combination with classification, in *Proceeding of HLT* (2007)
41. T.K. Ho, The random subspace method for constructing decision forests. IEEE Trans. PAMI **20**(8), 832–844 (1998)
42. HTK Toolkit, U.K. http://htk.eng.cam.ac
43. R. Hu, X. Li, Y. Zhao, Acoustic model training using greedy EM, in *Proceeding of ICASSP* (2005), pp. I697–700

44. B. Hutchinson, L. Deng, D. Yu, Tensor deep stacking networks. IEEE Trans. PAMI, **35**(8) (2013), 1944–1957
45. D. Jurafsky, J.H. Martin, *Speech and Language Processing*, 2nd ed., (Pearson-Prentice Hall, Englewood Cliffs, 2008)
46. B. Kingsbury, N. Morgan, Recognizing reverberant speech with Rasta-PLP, in *Proceeding of ICASSP* (1997), pp. 1259–1262
47. K. Kirchhoff, G.A. Fink, G. Sagerer, Combining acoustic and articulatory feature information for robust speech recognition. Speech Commun. **37**, 303–319 (2002)
48. A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, in *Advances in Neural Information Processing Systems*, ed. by G. Tesauro, D.S. Touretzky, T.K. Leen (MIT Press, Cambridge, 1995), pp. 231–238
49. L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Mach. Learn. **51**(2), 181–207 (2003)
50. L.I. Kuncheva, *Combining pattern classifiers – methods and algorithms* (Wiley, Hoboken, NJ, 2004)
51. A. Lazarevic, Z. Obradovic, Effective pruning of neural network classifier ensembles, in *Proceeding of ICNN* (2001), pp. 796–801
52. K. Livescu, E. Fosler-Lussier, F. Metze, Subword modeling for automatic speech recognition. IEEE SPM **29**(6), 44–57 (2012)
53. C. Ma, H.-K.J. Kuo, H. Soltan, X. Cui, U. Chaudhari, L. Mangu, C.-H. Lee, in *Proceeding of ICASSP* (2010), pp. 4394–4397
54. D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in *Proceeding of ICML* (1997), pp. 211–218
55. G. Martinez-Munoz, A. Suarez, Aggregation ordering in bagging, in *Proceeding of ICAIA* (2004), pp. 258–263
56. P. McMahon, P. McCourt, S. Vaseghi, Discriminative weighting of multi-resolution sub-band cepstral features for speech recognition, in *Proceeding of ICSLP* (1998), pp. 1055–1058
57. C. Meyer, H. Schramm, Boosting HMM acoustic models in large vocabulary speech recognition. Speech Commun. **48**, 532–548 (2006)
58. T. Mikolov, A. Deoras, S. Kombrink, L. Burget, J. Cernocky, Empirical evaluation and combination of advanced language modeling techniques, in *Proceeding of Interspeech* (2011)
59. D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, G. Zweig, FMPE: discriminatively trained features for speech recognition, in *Proceeding of ICASSP* (2005), pp. I-961–964
60. Y. Qian, J. Liu, Cross-lingual and ensemble MLPs strategies for low-resource speech recognition, in *Proceeding of Interspeech* (2012)
61. L. Rabiner, F. Juang, *Fundamentals of Speech Recognition* (Prentice Hall, Englewood Cliffs, 1993)
62. T. Robinson, M. Hochberg, S. Renals, The use of recurrent neural networks in continuous speech recognition, in *Automatic Speech and Speaker Recognition – Advanced Topics*, ed. by C.H. Lee, K.K. Paliwal, F.K. Soong (Kluwer Academic Publishers, Boston, 1995). Chapter 19
63. J.J. Rodriguz, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method. IEEE Trans. PAMI **28**(10), 1619–1630 (2006)
64. G. Saon, H. Soltau, Boosting systems for large vocabulary continuous speech recognition. Speech Commun. **54**(2), 212–218 (2012)
65. R.E. Schapire, The strength of weak learnability. Mach. Learn. **5**(2), 197–227 (1990)
66. H. Schwenk, Using boosting to improve a hybrid HMM/neural network speech recognition, in *Proceeding of ICASSP*, pp. 1009–1012 (1999)
67. T. Shinozaki, S. Furui, Spontaneous speech recognition using a massively parallel decoder, in *Proceeding of ICSLP* (2004), pp. 1705–1708
68. O. Siohan, B. Ramabhadran, B. Kingsbury, Constructing ensembles of ASR systems using randomized decision trees, in *Proceeding of ICASSP* (2005), pp. I-197–I-200
69. A. Strehl, J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)

70. E.K. Tang, P.N. Suganthan, X. Yao, An analysis of diversity measures. Mach. Learn. **65**(1), 247–271 (2006)
71. H. Tang, M. Hasegawa-Johnson, T. Huang, Toward robust learning of the Gaussian mixture state emission densities for hidden Markov models, in *Proceeding of ICASSP* (2010), pp. 2274–2277
72. K. Tumer, J. Ghosh, Analysis of decision boundaries in linearly combined neural classifiers. Pattern Recognit. **29**(2), 341–348 (1996)
73. G. Tur, L. Deng, D. Hakkani-Tur, X. He, Towards deeper understanding deep convex networks for semantic utterance classification, in *Proceeding of ICASSP* (2012)
74. N. Ueda, R. Nakano, Generalization error of ensemble estimators, in *Proceeding of ICNN* (1996), pp. 90–95
75. M.D. Wachter, M. Matton, K. Demuynck, P. Wambacq, P. Cools, D. Van Compernolle, Template-based continuous speech recognition. IEEE Trans. ASLP **15**(4), 1377–1390 (2007)
76. D.H. Wolpert, Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
77. S. Wu, B. Kingsbury, N. Mongan, S. Greenberg, Incorporating information from syllable-length time scales into automatic speech recognition, in *Proceeding of ICASSP* (1998), pp. 721–724
78. P. Xu, F. Jelinek, Random forest and the data sparseness problem in language modeling. Comput. Speech Lang. **21**, 105–152 (2007)
79. J. Xue, Y. Zhao, Random forests of phonetic decision trees for acoustic modeling in conversational speech recognition. IEEE Trans. ASLP **16**(3), 519–528 (2008)
80. R. Zhang, A. Rudnicky, Applying N-best list re-ranking to acoustic model combinations of boosting training, in *Proceeding of Interspeech* (2004a)
81. R. Zhang, A. Rudnicky, A frame level boosting training scheme for acoustic modeling, in *Proceeding of Interspeech* (2004b)
82. Y. Zhao, X. Zhang, R.-S. Hu, J. Xue, X. Li, L. Che, R. Hu, L. Schopp, An automatic captioning system for telemedicine, in *Proceeding of ICASSP* (2006), pp. I-957–I-960
83. Z.-H. Zhou, N. Li, Multi-information ensemble diversity, in *Proceeding of MCS* (2010), pp. 134–144
84. Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms* (CRC Press, Boca Raton, 2012)
85. Q. Zhu, A. Stolcke, B.Y. Chen, N. Morgan, Using MLP features in SRI's conversational speech recognition system, in *Proceedings of Interspeech* (2005), pp. 921–924
86. G. Zweig and M. Padmanabhan, Boosting Gaussian mixtures in an LVSCR system, Proc. ICASSP, pp. I-1527–I-1530 (2000)

Chapter 6
Deep Dynamic Models for Learning Hidden Representations of Speech Features

Li Deng and Roberto Togneri

Abstract Deep hierarchical structure with multiple layers of hidden space in human speech is intrinsically connected to its dynamic characteristics manifested in all levels of speech production and perception. The desire and an attempt to capitalize on a (superficial) understanding of this deep speech structure helped ignite the recent surge of interest in the deep learning approach to speech recognition and related applications, and a more thorough understanding of the deep structure of speech dynamics and the related computational representations is expected to further advance the research progress in speech technology. In this chapter, we first survey a series of studies on representing speech in a hidden space using dynamic systems and recurrent neural networks, emphasizing different ways of learning the model parameters and subsequently the hidden feature representations of time-varying speech data. We analyze and summarize this rich set of deep, dynamic speech models into two major categories: (1) top-down, generative models adopting localist representations of speech classes and features in the hidden space; and (2) bottom-up, discriminative models adopting distributed representations. With detailed examinations of and comparisons between these two types of models, we focus on the localist versus distributed representations as their respective hallmarks and defining characteristics. Future directions are discussed and analyzed about potential strategies to leverage the strengths of both the localist and distributed representations while overcoming their respective weaknesses, beyond blind integration of the two by using the generative model to pre-train the discriminative one as a popular method of training deep neural networks.

L. Deng (✉)
Microsoft Research, Redmond, WA 98034, USA
e-mail: deng@microsoft.com

R. Togneri
School of EE&C Engineering, The University of Western Australia, Crawley, WA 6009, Australia
e-mail: Roberto.Togneri@uwa.edu.au

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_6

6.1 Introduction

Before around 2010–2011, speech recognition technology had been dominated by a "shallow" architecture—hidden Markov models (HMMs) with each state characterized by a Gaussian mixture model (GMM). While significant technological success had been achieved using complex and carefully engineered variants of GMM-HMMs and acoustic features suitable for them, researchers long before that time had clearly realized that the next generation of speech recognition technology would require solutions to many new technical challenges under diversified deployment environments and that overcoming these challenges would likely require "deep" architectures that can at least functionally emulate the human speech system known to have dynamic and hierarchical structure in both production and perception [29, 36, 41, 96]. An attempt to incorporate a primitive level of understanding of this deep speech structure, initiated at the 2009 NIPS Workshop on Deep Learning for Speech Recognition and Related Applications [29], has helped create an impetus in the speech recognition community to pursue a deep representation learning approach based on the deep neural network (DNN) architecture, which was pioneered by the machine learning community only a few years earlier [52, 53] but rapidly evolved into the new state of the art in speech recognition with industry-wide adoption [16, 28, 29, 51, 59, 78, 93, 94, 108]. In the mean time, it has been recognized that the DNN approach (with its interface to the HMM) has not modeled speech dynamics properly. The deep and temporally recurrent neural network (RNN) has been developed to overcome this problem [13, 49], where the internal representation of dynamic speech features is discriminatively formed by feeding the low-level acoustic features into the hidden layer together with the recurrent hidden features from the past history. Even without stacking RNNs one on top of another as carried out in [49] or feeding DNN features as explored in [13], an RNN itself is a deep model since temporal unfolding of the RNN creates as many layers in the network as the length of the input speech utterance.

On the other hand, before the recent rise of deep learning for speech modeling and recognition, many earlier attempts had been made to develop computational architectures that are "deeper" than the conventional GMM-HMM architecture. One prominent class of such models are hidden dynamic models where the internal representation of dynamic speech features is generated probabilistically from the higher levels in the overall deep speech model hierarchy [12, 19, 23, 37, 65, 86, 92, 102]. Despite separate developments of the RNNs and of the hidden dynamic models, they share the same motivation—more realistically representing the dynamic structure of speech. Nevertheless, the different ways in which these two types of deep dynamic models are constructed endow them with distinct pros and cons. Investigations of the contrast between the two model types and the similarity to each other will yield insights into the strategies for developing new types of deep dynamic models with the hidden representations of speech features superior to the existing RNNs and hidden dynamic models. This forms the main motivation of this chapter.

In this chapter, we will focus on the most prominent contrast between the above two types of models in terms of the opposing localist and distributed representations adopted by the hidden layers in the models. In the distributed representation adopted by the RNNs, we cannot interpret the meaning of activity on a single unit or neuron in isolation. Rather, the meaning of the activity on any particular unit depends on the activities of other units. Using distributed representations, multiple concepts (i.e., phonological/linguistic symbols) can be represented at the same time on the same set of neuronal units by superimposing their patterns together. The strengths of distributed representations used by the RNN include robustness, representational and mapping efficiency, and the embedding of symbols into continuous-valued vector spaces which enable the use of powerful gradient-based learning methods. The localist representation adopted by the generative hidden dynamic models has very different properties. It offers very different advantages—easy to interpret, understand, diagnose, and easy to work with. Section 6.6 of this chapter will compare the two types of models, in terms of the localist versus distributed representations as well as other attributes, with respect to their strengths and weaknesses in detail. Based on such comparisons, Sect. 6.7 will discuss how to exploit the advantages of both types of models and of the representations they have adopted while circumventing their weaknesses. Before that and in Sects. 6.2–6.5, we will first provide a detailed review on major deep dynamic models in the literature relevant to our topic, focusing on the algorithms for learning model parameters from data and for computing the representations in the hidden spaces.

6.2 Generative Deep-Structured Speech Dynamics: Model Formulation

6.2.1 Generative Learning in Speech Recognition

In speech recognition, the most common generative learning approach is based on the GMM-HMM; e.g., [9, 30, 58, 88]. A GMM-HMM is a model that describes two dependent random processes, an observable process $\mathbf{x}_{1:T}$ and a hidden Markov process $y_{1:T}$. The observation x_t is assumed to be "generated" by the hidden state y_t according to a Gaussian mixture distribution. The GMM-HMM can be parameterized by $\lambda = (\pi, A, B)$; π is a vector of state prior probabilities; $A = (a_{i,j})$ is a state transition probability matrix; and $B = \{b_1, \ldots, b_n\}$ is a set where b_j represents the Gaussian mixture model of state j. The state is typically associated with a sub-segment of a phone in speech. One important innovation in speech recognition is the introduction of context-dependent states (e.g. [32, 88]), motivated by the desire to reduce output variability associated with each state, a common strategy for "detailed" generative modeling. A consequence of using context dependency is a vast expansion of the HMM state space, which, fortunately, can be controlled by regularization methods such as state tying.

The introduction of the HMM and the related statistical methods to speech recognition in mid 1970s [2,57] can be regarded the most significant paradigm shift in the field, as discussed in [3]. One major reason for this early success was due to the highly efficient maximum likelihood learning method invented about 10 years earlier [5]. This MLE method, often called the Baum–Welch algorithm, had been the principal way of training the HMM-based speech recognition systems until 2002, and is still one major step (among many) in training these systems nowadays. It is interesting to note that the Baum–Welch algorithm serves as one major motivating example for the later development of the more general Expectation-Maximization (EM) algorithm [17].

The goal of maximum likelihood learning is to minimize an empirical risk with respect to the joint likelihood loss (extended to sequential data), i.e.,

$$R_{\text{emp}}(f) = -\sum_i \ln p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}; \pi, A, B) \tag{6.1}$$

where $\mathbf{x}$ represents acoustic data, usually in form of a sequence feature vectors extracted at frame-level and $\mathbf{y}$ represents a sequence of linguistic units. It is crucial to apply some form of regularization to improve generalization. This leads to a practical training objective referred to as *accuracy-regularization* which takes the following general form:

$$J(f) = R_{\text{emp}}(f) + \gamma C(f) \tag{6.2}$$

where $C(f)$ is a regularizer that measures "complexity" of f, and γ is a tradeoff parameter. In large-vocabulary speech recognition systems, it is normally the case that word-level labels are provided, while state-level labels are latent. Moreover, in training HMM-based speech recognition systems, parameter tying is often used as a type of regularization [55]. For example, similar acoustic states of the triphones can share the same Gaussian mixture model. In this case, the $C(f)$ term is expressed by

$$C(f) = \prod_{(m,n)\in\mathcal{T}} \delta(b_m = b_n) \tag{6.3}$$

where $\mathcal{T}$ represents a set of tied state pairs.

The use of the generative model of HMMs, including the most popular Gaussian-mixture HMM, for representing the (piece-wise stationary) dynamic speech pattern and the use of MLE for training the tied HMM parameters constitutes one of the most prominent and successful examples of generative learning in speech recognition. This success was firmly established by the speech recognition community, and has been widely spread to the machine learning and related communities; in fact, the HMM has become a standard tool not only in speech recognition but also in machine learning and their related fields such as bioinformatics and natural language processing. For many machine learning as well as speech recognition researchers, the success of the HMM in speech recognition is a bit surprising due to the well-known weaknesses of the HMM.

Another clear success of the generative learning paradigm in speech recognition is the use of the GMM-HMM as prior "knowledge" within the Bayesian framework for environment-robust speech recognition. The main idea is as follows. When the speech signal, to be recognized, is mixed with noise or another non-intended speaker, the observation is a combination of the signal of interest and interference of no interest, both unknown. Without prior information, the recovery of the speech of interest and its recognition would be ill defined and subject to gross errors. Exploiting generative models of GMM-HMMs, or often simpler GMMs, as Bayesian priors for "clean" speech overcomes the ill-posed problem. Further, the generative approach allows probabilistic construction of the model for the relationship between the noisy speech observation, clean speech, and interference, which is typically nonlinear when the log-domain features are used. A set of generative learning approaches in speech recognition following this philosophy are variably called "parallel model combination" [45], vector Taylor series (VTS) method [1,26], and Algonquin [44]. Notably, the comprehensive application of such a generative learning paradigm for single-channel multitalker speech recognition is reported and reviewed in [89], where the authors apply successfully a number of well established ML methods including loopy belief propagation and structured mean-field approximation. Using this generative learning scheme, speech recognition accuracy with loud interfering speakers is shown to exceed human performance.

Despite some success of GMM-HMMs in speech recognition, their weaknesses, such as the conditional independence assumption, have been well known for speech recognition applications [3,4]. Since the early 1990s, speech recognition researchers have begun the development of statistical models that capture the dynamic properties of speech in the temporal dimension more faithfully than HMMs. This class of beyond-HMM models have been variably called the stochastic segment model [81, 82], trended or nonstationary-state HMM [18, 24], trajectory segmental model [54, 81], trajectory HMMs [63, 111, 112], stochastic trajectory models [47], hidden dynamic models [12,19,23,37,65,86,92,102], buried Markov models [8], structured speech model [40], and the hidden trajectory model [39] depending on different "prior knowledge" applied to the temporal structure of speech and on various simplifying assumptions to facilitate the model implementation. Common to all these beyond-HMM models is some temporal trajectory structure built into the models, hence trajectory models. Based on the nature of such a structure, we can classify these models into two main categories. In the first category are the models focusing on a temporal correlation structure at the "surface" acoustic level. The second category consists of hidden dynamics, where the underlying speech production mechanisms are exploited as the Bayesian prior to represent the temporal structure that accounts for the observed speech pattern. When the mapping from the hidden dynamic layer to the observation layer is limited to linear (and deterministic), then the generative hidden dynamic models in the second category reduces to the first category.

The temporal span of the generative trajectory models in both categories above is controlled by a sequence of linguistic labels, which segment the full sentence into multiple regions from left to right; hence segment models.

In a general form, the trajectory/segment models with hidden dynamics make use of the switching state space formulation. They use temporal recursion to define the hidden dynamics, $\mathbf{z}(k)$, which may correspond to articulatory movement during human speech production. Each discrete region or segment, s, of such dynamics is characterized by the s-dependent parameter set $\boldsymbol{\Lambda}_s$, with the "state noise" denoted by $\mathbf{w}_s(k)$. The memory-less nonlinear mapping function is exploited to link the hidden dynamic vector $\mathbf{z}(k)$ to the observed acoustic feature vector $\mathbf{o}(k)$, with the "observation noise" denoted by $\mathbf{v}_s(k)$, and parameterized also by segment-dependent parameters. The combined "state equation" (6.4) and "observation equation" (6.5) below form a general switching nonlinear dynamic system model:

$$\mathbf{z}(k+1) = \mathbf{g}_k[\mathbf{z}(k), \boldsymbol{\Lambda}_s] + \mathbf{w}_s(k) \tag{6.4}$$

$$\mathbf{o}(k') = \mathbf{h}_{k'}[\mathbf{z}(k'), \boldsymbol{\Omega}_{s'}] + \mathbf{v}_{s'}(k'). \tag{6.5}$$

where subscripts k and k' indicate that the functions $\mathbf{g}[.]$ and $\mathbf{h}[.]$ are time varying and may be asynchronous with each other. s or s' denotes the dynamic region correlated with phonetic categories.

The model expressed by (6.4) and (6.5) is not only dynamic, but also deep since there is a hierarchy of information flow from discrete linguistic symbols s to the hidden dynamic vector $\mathbf{z}(k)$ and then to the observed vectors $\mathbf{o}(k)$. We call this type of model a generative deep-structured dynamic model. Being "generative" here means that the model provides a causal relationship from the (top) linguistic labels to intermediate and then to the (bottom) observed acoustic variables. This distinguishes from the "discriminative" deep-structured models where the information flow starts from the (bottom) observed acoustic variables to the intermediate representations and then to the (top) linguistic labels.

There have been several studies on switching nonlinear state space models for speech recognition, both theoretical [21, 37] and experimental [12, 61, 65, 86]. The specific forms of the functions of $\mathbf{g}_k[\mathbf{z}(k), \boldsymbol{\Lambda}_s]$ and $\mathbf{h}_{k'}[\mathbf{z}(k'), \boldsymbol{\Omega}_{s'}]$ and their parameterization are determined by prior knowledge based on the current understanding of the nature of the temporal dimension in speech. In particular, state equation (6.4) takes into account the temporal elasticity in spontaneous speech and its correlation with the "spatial" properties in hidden speech dynamics such as articulatory positions or vocal tract resonance frequencies; see [23] for a comprehensive review of this body of work.

When nonlinear functions of $\mathbf{g}_k[\mathbf{z}(k), \boldsymbol{\Lambda}_s]$ and $\mathbf{h}_{k'}[\mathbf{z}(k'), \boldsymbol{\Omega}_{s'}]$ in (6.4) and (6.5) are reduced to linear functions (and when synchrony between the two equations are eliminated), the switching nonlinear dynamic system model is reduced to its linear counterpart, the switching linear dynamic system. It can be viewed as a hybrid of standard HMMs and linear dynamical systems, with a general mathematical description of

$$\mathbf{z}(k+1) = \mathbf{A}_s\mathbf{z}(k) + \mathbf{B}_s\mathbf{w}_s(k) \tag{6.6}$$

$$\mathbf{o}(k) = \mathbf{C}_s\mathbf{z}(k) + \mathbf{v}_s(k). \tag{6.7}$$

There has also been an interesting set of work on the switching linear dynamic system applied to speech recognition. The early set of studies have been carefully reviewed in [81] for generative speech modeling and for its speech recognition applications. The studies reported in [42, 72] further applied this system model to noise-robust speech recognition and explored several approximate inference techniques, overcoming intractability in decoding and parameter learning. The study reported in [91] applied another approximate inference technique, a special type of Gibbs sampling commonly used in machine learning, to a speech recognition problem.

During the development of trajectory/segment models for speech recognition, a number of machine learning techniques invented originally in non-speech recognition communities, e.g. variational learning [61], pseudo-Bayesian [42, 65], Kalman filtering [81], extended Kalman filtering [23, 37, 101], Gibbs sampling [91], orthogonal polynomial regression [24], etc., have been usefully applied with modifications and improvement to suit the speech-specific properties and speech recognition applications. However, the success has mostly been limited to small-scale tasks. We can identify four main sources of difficulty (as well as new opportunities) in successful applications of trajectory/segment models to large-scale speech recognition. First, scientific knowledge on the precise nature of the underlying articulatory speech dynamics and its deeper articulatory control mechanisms is far from complete. Coupled with the need for efficient computation in training and decoding for speech recognition applications, such knowledge has been forced to be again simplified, reducing the modeling power and precision further. Second, most of the work in this area has been placed within the generative learning setting, having a goal of providing parsimonious accounts (with small parameter sets) for speech variations due to contextual factors and co-articulation. In contrast, the recent joint development of deep learning by both ML and speech recognition communities, which we will review in Sect. 6.6, combines generative and discriminative learning paradigms and makes use of massive instead of parsimonious parameters. There is a huge potential for synergy of research here. Third, although structural ML learning of switching dynamic systems via Bayesian nonparametrics has been maturing and producing successful applications in a number of ML and signal processing tasks (e.g. the tutorial paper [43]), it has not entered mainstream speech recognition; only isolated studies have been reported on using Bayesian nonparametrics for modeling aspects of speech dynamics [83] and for language modeling [14]. Finally, most of the trajectory/segment models developed by the speech recognition community have focused on only isolated aspects of speech dynamics rooted in deep human production mechanisms, and have been constructed using relatively simple and largely standard forms of dynamic systems.

In the remainder of this section, we will review two special cases of the general dynamic models of speech represented by (6.4)–(6.7) with hidden structure. These models are considered to be "deep", in that the hidden structure is modeled as an intermediate information processing stage connecting the linguistic information to the observable acoustics.

6.2.2 A Hidden Dynamic Model with Nonlinear Observation Equation

Let us consider in detail the hidden dynamic model (HDM) using the extended Kalman filter [102]. The hidden dynamics is chosen to be the vocal-tract-resonances (VTRs), which are closely related to the smooth and target-oriented movement of the articulators. The first component of the HDM, also called the state equation, is a target-directed, continuously-valued (hidden) Markov process that is used to describe the hidden VTR dynamics according to:

$$\mathbf{z}(k+1) = \mathbf{\Phi}_s \mathbf{z}(k) + (\mathbf{I}_m - \mathbf{\Phi}_s)\mathbf{T}_s + \mathbf{w}(k) \tag{6.8}$$

where $\mathbf{z}(k)$ is the $m \times 1$ VTR state vector, $\mathbf{T}_s$ is the $m \times 1$ phone target vector parameter and $\mathbf{\Phi}_s$ is the $m \times m$ diagonal "time-constant" matrix parameter associated with the phone regime s. The phone regime is used to describe the segment of speech that is attributed to the phone identified by the model pair $(\mathbf{\Phi}_s, \mathbf{T}_s)$. The process noise $\mathbf{w}(k)$ is an i.i.d, zero-mean, Gaussian process with covariance $\mathbf{Q}$. The target-directed nature of the process is evident by noting that $\mathbf{z}(k) \rightarrow \mathbf{T}_s$ as $k \rightarrow \infty$ independent of the initial value of the state.

The second component of the HDM is the observation equation used to describe the static mapping from the lower dimensional hidden VTR state vector (typically $m = 3$ for the first three VTR resonances) to the higher dimensional observable acoustic feature vector. The general form of this mapping assumes a static, multivariate nonlinear mapping function as follows:

$$\mathbf{o}(k) = h_r(\mathbf{z}(k)) + \mathbf{v}(k). \tag{6.9}$$

where the $n \times 1$ acoustic observation $\mathbf{o}(k)$ is the set of acoustic feature vectors for frame k (the usual Mel-frequency cepstral co-efficient (MFCC) features with $n = 12$), and $h_r(\mathbf{z}(k))$ is the $n \times m$ static, non-linear mapping function on the state vector $\mathbf{z}(k)$ associated with the manner of articulation r. The manner of articulation describes how the phone is articulated to produce the acoustic observations arising from the speech production process and will usually be different for the different broad phonetic classes (e.g. vowels, voiced stops, etc.). The observation noise $\mathbf{v}(k)$ is an i.i.d, zero-mean, Gaussian process with covariance $\mathbf{R}$. The multivariate mapping function $h_r(\mathbf{z}(k))$ is implemented by a m-J-n feedforward multi-layer perceptron (MLP) with J hidden nodes, a linear activation function on the output layer, and the antisymmetric hyperbolic tangent function on the hidden layer. There is a unique MLP network for each distinct r.

The switching state behaviour of this model is represented by an M-state discrete-time random sequence, where $s \equiv s(k) \in [1, 2 \ldots, M]$ is a random variable that takes on one of the M possible "phone" regimes (or states) at time k. An additional R-state discrete-time random sequence also exists where $r \equiv r(k) \in [1, 2, \ldots R]$ is a random variable that takes on one of the R possible manner of

articulation states at time k. In practice both sequences are unknown and need to be estimated, both when training the model (i.e. estimating the parameters) and testing (i.e. using the model to rescore or decode an unknown observation sequence).

An important property of this model is the continuity of the hidden state variable $\mathbf{z}(k)$ across phone regimes. That is, $\mathbf{z}(k)$ at the start of segment $l+1$ is set to the value computed at the end of segment l. This provides a long-span continuity constraint across adjacent phone regimes that structurally models the inherent context dependencies and coarticulatory effects [35].

An important concern is the specific modeling of the state dynamic and observation process. The target-directed state dynamic is reasonable but requires knowledge of the per-phone target and time-constant values. If these are not known these have to be jointly estimated. The non-linear mapping from the state vector to observation vector is more problematic as the MLP weights also have to be estimated and this creates a system with too many degrees of freedom. Possible solutions to do this have included: using prior VTR measurement data to independently train the MLP [102], using a more simple linear mapping [61], or restricting to observation features like LPC cepstra which permit an analytical mapping with the VTR resonances [31]. Finally we also assume that the phone sequence or segmentation of model regimes, $s(k)$, is known in advance, which, in practice, requires training on phonetically transcribed speech corpora.

6.2.3 *A Linear Hidden Dynamic Model Amenable to Variational EM Training*

An alternative approach to implementing the hidden dynamic model is to reformulate it in the context of a segmental switching state space model and to apply the variational EM algorithm to learn the model parameters. The state equation and observation equation in this reformulated model, as described in [61], are

$$\mathbf{x}_n = \mathbf{A}_s \mathbf{x}_{n-1} + (\mathbf{I} - \mathbf{A}_s)\mathbf{u}_s + \mathbf{w}, \tag{6.10}$$

$$\mathbf{y}_n = \mathbf{C}_s \mathbf{x}_n + \mathbf{c}_s + \mathbf{v}, \tag{6.11}$$

where n and s are frame number and phone index respectively, $\mathbf{x}$ is the hidden dynamics and $\mathbf{y}$ is the acoustic feature vector (such as MFCC). The hidden dynamics are chosen to be the vocal-tract-resonances (VTRs). The state equation (6.10) is a linear dynamic equation with phone dependent system matrix $\mathbf{A}_s$ and target vector $\mathbf{u}_s$ and with built-in continuity constraints across the phone boundaries. The observation equation (6.11) represents a phone-dependent VTR-to-acoustic linear mapping. The choice of linear mapping is mainly due to the difficulty of algorithm development. The resulting algorithm can also be generalized to mixtures of linear mappings and piece-wise linear mappings within a phone. Gaussian white noises

$\mathbf{w}_n$ and $\mathbf{v}_n$ are added to both the state and observation equations to make the model probabilistic. Similar models have been proposed and used previously [35, 62].

To facilitate algorithm development, the model is also expressed in terms of probability distributions:

$$\begin{aligned} p(s_n = s \mid s_{n-1} = s') &= \pi_{ss'}, \\ p(\mathbf{x}_n \mid s_n = s, \mathbf{x}_{n-1}) &= \mathcal{N}(\mathbf{x}_n \mid \mathbf{A}_s x_{n-1} + \mathbf{a}_s, \mathbf{B}_s), \\ p(\mathbf{y}_n \mid s_n = s, \mathbf{x}_n) &= \mathcal{N}(\mathbf{y}_n \mid \mathbf{C}_s \mathbf{x}_n + \mathbf{c}_s, \mathbf{D}_s), \end{aligned} \tag{6.12}$$

where $\pi_{ss'}$ is the phone transition probability matrix, $\mathbf{a}_s = (\mathbf{I} - \mathbf{A}_s)\mathbf{u}_s$ and $\mathcal{N}$ denotes a Gaussian distribution with mean and precision matrix (inverse of the covariance matrix) as the parameters. The joint distribution over the entire time sequence is given by

$$p(\mathbf{y}_{1:N}, \mathbf{x}_{1:N}, s_{1:N}) = \prod_n p(\mathbf{y}_n | s_n, \mathbf{x}_n) p(\mathbf{x}_n | s_n, \mathbf{x}_{n-1}) p(s_n | s_{n-1}). \tag{6.13}$$

The conditional independence relations of the model can be seen more clearly from a graphic form (Bayesian network) as shown in Fig. 6.1.

There are a few issues to be solved before any estimation or learning algorithms can be applied to speech, and they are discussed here:

1. Parameter initialization: It is important to initialize the parameters appropriately for an iterative local optimization procedure such as EM. The HDM enjoys the benefit of being closely related to speech-specific knowledge and some key parameters, especially the phone targets, can be reliably initialized from a formant synthesizer. Due to the small number of total parameters, others can be easily initialized by a small amount of hand-labeled VTR data.

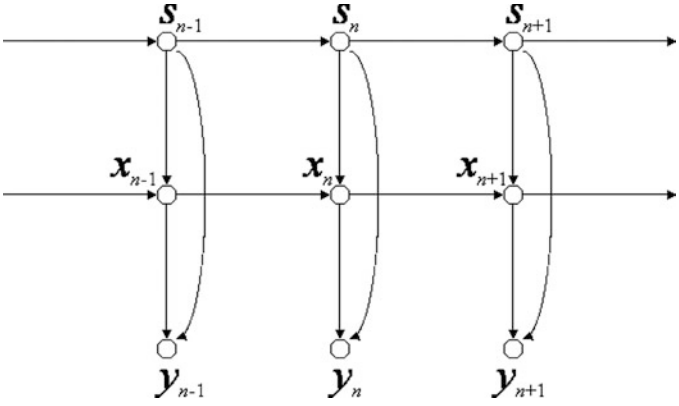

Fig. 6.1 HDM represented as a Bayesian network

2. Segmental constraint: The probabilistic form of the model allows phone transitions to occur at each frame, which is undesirable for speech. In training, we construct a series of time-varying transition matrices $\pi_{ss'}$ based on the given phonetic transcript (or one created from a lexicon if only word transcripts are given) and some initial segmentation to impose the segmental constraint and force the discrete-state component of the model to be consistent with the phonetic transcript. Such an approach also greatly reduces the number of possible phones s that have to be summed at each time step.

6.3 Generative Deep-Structured Speech Dynamics: Model Estimation

6.3.1 Learning a Hidden Dynamic Model Using the Extended Kalman Filter

The estimation problem that we investigate is as follows. Given multiple sets of observation sequences, $\mathbf{o}(k)$, for each distinct phone regime, we seek to determine the optimal estimates for the unknown values of the state-equation parameters $\mathbf{\Phi}$ and $\mathbf{T}$, and the observation-equation parameters, $\mathbf{W}$, which is the MLP weight vector of the nonlinear mapping function $h(\mathbf{z}(k))$. For clarity of notation we will drop the s and r subscripts since it is understood the estimation equations only apply for observations taken over a particular phone regime segment.

The expectation-maximization (EM) algorithm is a widely used algorithm for the estimation of the parameters in general state-space models and in the current research on the HDM [34, 35]. The EM algorithm provides new estimates of the parameters after the set of all available N observation vectors have been presented. The EM algorithm can be considered a batch or off-line estimation method most suited to applications where all of the data is available. We now present the EM algorithm for the specific type of model given by (6.8) and (6.9) following [101, 102].

6.3.1.1 E-Step

For a sequence of N observation vectors, the E-step involves computation of the conditional expectation of the log joint likelihood between $\mathbf{Z} = \{\mathbf{z}(0), \mathbf{z}(1), \ldots, \mathbf{z}(N)\}$ and $\mathbf{O} = \{\mathbf{o}(0), \mathbf{o}(1), \ldots, \mathbf{o}(N)\}$ given the observation $\mathbf{O}$ and parameter set $\overline{\Theta}$ estimated at the previous step, that is:

$$
\begin{aligned}
Q(\Theta|\overline{\Theta}) &= E\{\log L(\mathbf{Z},\mathbf{O}|\Theta)|\mathbf{O},\overline{\Theta}) \\
&= -\frac{1}{2}\sum_{k=0}^{N-1} E_N[\mathbf{e}_{k1}^T\mathbf{Q}^{-1}\mathbf{e}_{k1}|\mathbf{O},\overline{\Theta}] \\
&\quad -\frac{1}{2}\sum_{k=0}^{N-1} E_N[\mathbf{e}_{k2}^T\mathbf{R}^{-1}\mathbf{e}_{k2}|\mathbf{O},\overline{\Theta}] + const
\end{aligned} \tag{6.14}
$$

where $\mathbf{e}_{k1} = [\mathbf{z}(k+1) - \mathbf{\Phi}\mathbf{z}(k) - (\mathbf{I}-\mathbf{\Phi})\mathbf{T}]$ and $\mathbf{e}_{k2} = [\mathbf{o}(k) - h(\mathbf{z}(k))]$ and E_N denotes the expectation based on N samples. The standard EKF smoother is used to provide estimates of the hidden dynamic variable, $\mathbf{z}(k) \equiv \hat{\mathbf{z}}(k|N) = E_N[\mathbf{z}(k)|\mathbf{O},\overline{\Theta}]$. The Jacobian matrix for the $n \times m$ non-linear mapping function $h(\mathbf{z}(k))$ used in the EKF recursion is given by:

$$
\begin{aligned}
H_z^{ji}[\hat{\mathbf{z}}(k+1|k)] &= \left[\frac{\partial o_j(k+1)}{\partial \hat{z}_i(k+1|k)}\right] \\
&= \left[\sum_{h=1}^{J} W_{2j}^h g'(\mathbf{W}_{1h}^T\hat{\mathbf{z}}(k+1|k))W_{1h}^i\right]
\end{aligned} \tag{6.15}
$$

where $o_j(k)$ is the jth component of the observation vector at time k, $\hat{z}_i(k+1|k)$ is the ith component of the predicted state vector $\hat{z}(k+1|k)$ at time k, W_{lh}^i is the ith component of the MLP weight vector, $\mathbf{W}_{lh}$, of node h in layer l (layer 1 is the hidden layer and layer 2 is the output layer), J is the number of nodes in the hidden layer and $g'(x)$ is the derivative of the activation function in the hidden layer.

It should be noted that the continuity condition on $\hat{\mathbf{z}}(k)$ is also applied to the EKF error covariance.

6.3.1.2 M-Step

In the M-step the Q function in (6.14) is maximised with respect to the parameter set $\Theta = (\mathbf{T},\mathbf{\Phi},\mathbf{W})$. We consider the first summation involving the parameters $\mathbf{T}$ and $\mathbf{\Phi}$:

$$
Q_1(\mathbf{Z},\mathbf{O},\Theta) = \sum_{k=0}^{N-1} E_N[\mathbf{e}_{k1}^T\mathbf{Q}^{-1}\mathbf{e}_{k1}|O,\overline{\Theta}] \tag{6.16}
$$

Minimisation of Q_1, which implies maximisation of Q, proceeds by setting the partial derivatives with respect to $\mathbf{T}$ and $\mathbf{\Phi}$ to zero, that is:

$$
\frac{\partial Q_1}{\partial \mathbf{\Phi}} \propto \sum_{k=0}^{N-1} E_N\{[\mathbf{z}(k+1) - \mathbf{\Phi}\mathbf{z}(k) - (\mathbf{I}-\mathbf{\Phi})\mathbf{T}][\mathbf{T}-\mathbf{z}(k)]^T|\mathbf{O},\overline{\Theta}\} = 0 \tag{6.17}
$$

$$\frac{\partial Q_1}{\partial \mathbf{T}} \propto \sum_{k=0}^{N-1} E_N\{[\mathbf{z}(k+1) - \mathbf{\Phi}\mathbf{z}(k) - (\mathbf{I}-\mathbf{\Phi})\mathbf{T}]|\mathbf{O}, \overline{\Theta}\} = 0 \tag{6.18}$$

The resulting equations to be solved are nonlinear high-order equations in terms of $\mathbf{\Phi}$ and $\mathbf{T}$:

$$N\mathbf{\Phi}\mathbf{T}\mathbf{T}^T - \mathbf{\Phi}\mathbf{T}\mathbf{A}^T - \mathbf{\Phi}\mathbf{A}\mathbf{T}^T - N\mathbf{T}\mathbf{T}^T + \mathbf{T}\mathbf{A}^T + \mathbf{B}\mathbf{T}^T + \mathbf{\Phi}\mathbf{C} - \mathbf{D} = 0 \tag{6.19}$$

$$\mathbf{B} - \mathbf{\Phi}\mathbf{A} - N\mathbf{T} + N\mathbf{\Phi}\mathbf{T} = 0 \tag{6.20}$$

where:

$$\mathbf{A} = \sum_{k=0}^{N-1} E_N[\mathbf{z}(k)|\mathbf{O}, \overline{\Theta}], \quad \mathbf{C} = \sum_{k=0}^{N-1} E_N[\mathbf{z}(k)\mathbf{z}(k)^T|\mathbf{O}, \overline{\Theta}] \tag{6.21}$$

$$\mathbf{B} = \sum_{k=0}^{N-1} E_N[\mathbf{z}(k+1)|\mathbf{O}, \overline{\Theta}], \quad \mathbf{D} = \sum_{k=0}^{N-1} E_N[\mathbf{z}(k+1)\mathbf{z}(k)^T|\mathbf{O}, \overline{\Theta}] \tag{6.22}$$

are the relevant sufficient statistics that are computed by the EKF smoother during the E-step. By simplifying (6.19) and (6.20) we can first form:

$$\hat{\mathbf{\Phi}} = \mathbf{X}\mathbf{Y}^{-1} \tag{6.23}$$

where $\hat{\mathbf{\Phi}}$ is the estimate of the system matrix, and then:

$$\hat{\mathbf{T}} = \frac{1}{N}(\mathbf{I} - \hat{\mathbf{\Phi}})^{-1}(\mathbf{B} - \hat{\mathbf{\Phi}}\mathbf{A}) \tag{6.24}$$

where $\hat{\mathbf{T}}$ is the estimate of the target vector.

We now consider the second summation of the Q function in (6.14) involving the parameter $\mathbf{W}$:

$$Q_2(\mathbf{Z}, \mathbf{O}, \Theta) = \sum_{k=0}^{N-1} E_N[\mathbf{e}_{k2}^T \mathbf{R}^{-1} \mathbf{e}_{k2}|\mathbf{O}, \overline{\Theta}] \tag{6.25}$$

Minimisation of Q_2, which leads to maximisation of Q, proceeds by setting the partial derivatives with respect to $\mathbf{W}$ to zero, that is:

$$\frac{\partial Q_2}{\partial \mathbf{W}} \propto \sum_{k=0}^{N-1} E_N[\frac{\partial}{\partial \mathbf{W}}\{[\mathbf{o}(k) - h(\mathbf{z}(k)]^2\}|\mathbf{O}, \overline{\Theta}] = 0 \tag{6.26}$$

That is, Q_2 is minimised when the error signal, $\mathbf{e}_{k2} = \mathbf{o}(k) - h(\mathbf{z}(k))$, is minimised. Since the multi-variate mapping function is a feedforward MLP network, then the standard back-propagation is used with $\hat{\mathbf{z}}(k|N)$ as the input and $\mathbf{o}(k)$ as the desired output to provide estimates of the MLP weights, $\mathbf{W}$.

6.3.2 Learning a Hidden Dynamic Model Using Variational EM

6.3.2.1 Model Inference and Learning

For the system described by (6.10)–(6.13) inference refers to the calculation of posterior distribution $p(s_{1:N}, \mathbf{x}_{1:N} \mid \mathbf{y}_{1:N})$ given all model parameters, while learning refers to the estimation of model parameters $\Theta = \{\mathbf{A}_{1:S}, \mathbf{a}_{1:S}, \mathbf{B}_{1:S}, \mathbf{C}_{1:S}, \mathbf{c}_{1:S}, \mathbf{D}_{1:S}\}$ given the complete distribution, usually in a maximum likelihood (ML) sense. Under this EM framework, inference is the E step and learning is the M step. In this model, however, the posterior turns out to be a Gaussian mixture whose number of components is exponential in the number of states (or phones) and frames, and is therefore computationally intractable. Here we develop two approximations, called GMM and HMM posteriors, respectively, based on *variational* techniques. The idea is to choose the approximate posterior $q(s_{1:N}, \mathbf{x}_{1:N} \mid \mathbf{y}_{1:N})$ with a sensible and tractable structure and optimize it by minimizing its Kullback-Liebler (KL) distance to the exact posterior. It turns out that this optimization can be performed efficiently without having to compute the exact (but intractable) posterior.

It is necessary to say a few words about previous approaches and other related work in the literature before presenting the current one. Most of our previous algorithms are developed under the assumption of hard phone boundaries which are either known or estimated separately by some heuristic methods [65], and the intractable exact posterior is approximated by a single Gaussian. This is also true for most of the work in a broad range of literatures for switching state space models. In contrast, the approach presented here uses soft phone assignments that are estimated under a unified EM framework as in [46, 85], but unlike [46, 85], our approximation doesn't factorize s from $\mathbf{x}$ and results in a multimodal posterior over $\mathbf{x}$ instead of a unimodal one, which is justifiably more suitable for speech applications.

6.3.2.2 The GMM Posterior

Under this approximation q is restricted to be:

$$q(s_{1:N}, \mathbf{x}_{1:N}) = \prod_n q(\mathbf{x}_n \mid s_n) q(s_n), \tag{6.27}$$

where from now on the dependence of the q's on the data $\mathbf{y}$ is omitted but always implied.

Minimizing the KL divergence between q and p is equivalent to maximizing the following functional $\mathcal{F}$,

$$\mathcal{F}[q] = \sum_{s_{1:N}} \int d\mathbf{x}_{1:N}\; q(s_{1:N}, \mathbf{x}_{1:N}) \cdot \left[\log p(\mathbf{y}_{1:N}, \mathbf{x}_{1:N}, s_{1:N}) - \log q(s_{1:N}, \mathbf{x}_{1:N})\right], \tag{6.28}$$

which is also a lower bound of the likelihood function and will be subsequently used as the objective function in the learning (M) step.

By taking *calculus of variation* to optimize $\mathcal{F}$ w.r.t. $q(\mathbf{x}_n|s_n)$ and $q(s_n)$, it turns out that each component $q(\mathbf{x}_n|s_n)$ follows a Gaussian distribution, i.e.,

$$q(\mathbf{x}_n \mid s_n = s) = \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\rho}_{s,n}, \boldsymbol{\Gamma}_{s,n}), \tag{6.29}$$

and the parameters $\boldsymbol{\rho}_{s,n}$ and $\boldsymbol{\Gamma}_{s,n}$ are given by

$$\boldsymbol{\Gamma}_{s,n} = \mathbf{C}_s^T \mathbf{D}_s \mathbf{C}_s + \mathbf{B}_s + \sum_{s'} \gamma_{s',n+1} \mathbf{A}_{s'}^T \mathbf{B}_{s'} \mathbf{A}_{s'}, \tag{6.30}$$

$$\begin{aligned} \boldsymbol{\Gamma}_{s,n}\boldsymbol{\rho}_{s,n} = \mathbf{B}_s(\mathbf{A}_s \sum_{s'} \gamma_{s',n-1}\boldsymbol{\rho}_{s',n-1} + \mathbf{a}_s) \\ + \sum_{s'} \gamma_{s',n+1}\mathbf{A}_{s'}^T \mathbf{B}_{s'}(\boldsymbol{\rho}_{s',n+1} - \mathbf{a}_{s'}) \\ + \mathbf{C}_s^T \mathbf{D}_s(\mathbf{y}_n - \mathbf{c}_s), \end{aligned} \tag{6.31}$$

where $\gamma_{s,n} = q(s_n = s)$ and is computed from

$$\begin{aligned} \log \gamma_{s,n} = f_1(\boldsymbol{\rho}_{s,n}, \boldsymbol{\Gamma}_{s,n}, \Theta) + f_2(\boldsymbol{\rho}_{s',n-1}, \boldsymbol{\Gamma}_{s',n-1}, \Theta) \\ + f_3(\boldsymbol{\rho}_{s',n+1}, \boldsymbol{\Gamma}_{s',n+1}, \Theta). \end{aligned} \tag{6.32}$$

and the f's denote linear functions whose expressions are too lengthy to be written down here. Equations (6.30) and (6.31) are coupled linear equations given model parameters Θ and γ's and can be solved efficiently by sparse matrix techniques. Equation (6.32) is a nonlinear equation by itself and has to be solved by iteration. Equations (6.30)–(6.32) constitute the inference or E step of the algorithm and have to be solved iteratively all together after some proper initializations.

Model learning involves taking derivatives of $\mathcal{F}$ w.r.t. all the model parameters and setting them to zero. This results in a set of linear equations which can be solved easily. Since this step is standard in all EM approaches with no special difficulties, the detailed equations are omitted.

6.3.2.3 The HMM Posterior

Under this approximation q is taken to be

$$q(s_{1:N}, \mathbf{x}_{1:N}) = \prod_{n=1}^{N} q(\mathbf{x}_n \mid s_n) \cdot \prod_{n=2}^{N} q(s_n \mid s_{n-1}) \cdot q(s_1). \tag{6.33}$$

First we define two posterior transition probabilities:

$$\eta_{s's,n} = q(s_n = s \mid s_{n-1} = s'),$$
$$\bar{\eta}_{s's,n} = q(s_n = s \mid s_{n+1} = s') = \frac{\eta_{s's,n+1}\gamma_{s,n}}{\gamma_{s',n+1}}, \tag{6.34}$$

where γ is the same as in the previous section. It turns out that each $q(\mathbf{x}_n|s_n)$ is again a Gaussian distribution, and $\boldsymbol{\rho}_{s,n}$ and $\boldsymbol{\Gamma}_{s,n}$ are given by coupled linear equations having the same form as (6.30) and (6.31), except that the γ's are replaced by η's and $\bar{\eta}$'s. These equations can again be solved by sparse matrix techniques. The γ's and η's themselves can be solved by the following efficient backward-forward procedure given the model parameters and all the $\boldsymbol{\rho}$'s and $\boldsymbol{\Gamma}$'s.

1. Initialize: $z_{s,N+1} = 1$ for all s.
2. Backward pass: for $n = N, \cdots, 2$

$$z_{s,n} = \sum_{s'} \exp(f_{ss',n}) z_{s',n+1} \ ,$$
$$\eta_{ss',n} = \frac{1}{z_{s,n}} \exp(f_{ss',n}) z_{s',n+1} \ . \tag{6.35}$$

3. For $n = 1$:

$$z_1 = \sum_{s} \exp(f_{s,1}) z_{s,2} \ ,$$
$$\gamma_{s,1} = \frac{1}{z_1} \exp(f_{s,1}) z_{s,2} \ . \tag{6.36}$$

4. Forward pass: for $n = 2, \cdots, N$

$$\gamma_{s,n} = \sum_{s'} \eta_{s's,n} \gamma_{s',n-1} \ . \tag{6.37}$$

Again, f's are functions of the $\boldsymbol{\rho}$'s, $\boldsymbol{\Gamma}$'s and model parameters whose expressions are too lengthy to be given here. Also remember that the complete E step still has to iterate between the calculation of $q(\mathbf{x}_n \mid s_n)$ and $q(s_n \mid s_{n-1})$. The model learning is quite similar to the GMM case and the detailed equations are omitted.

There are a number of important issues to be addressed when using the above algorithms for speech:

1. Hidden dynamics recovery: It is both informative (especially for debugging) and desirable to recover the hidden VTR, and it is calculated by:

$$\hat{x}_n = \sum_{s} \gamma_{s,n} \boldsymbol{\rho}_{s,n} \tag{6.38}$$

for both the GMM and HMM posterior assumptions.

2. Recognition strategy: Here we seek the most likely phone sequence given a sequence of observations. For the GMM case, this is simply accomplished by choosing the maximum γ at each frame; while for the HMM posterior we need to perform Viterbi decoding by using γ and η, e.g., the initialization and induction equation for the scoring are:

$$V_1(s) = \gamma_{s,1}, \; V_n(s') = \max_{1 \le s \le S} [V_{n-1}(s)\eta_{ss',n}]\, \gamma_{s',n}. \tag{6.39}$$

It is highly desirable to incorporate segmental (or minimal duration) constraint and language weighting in the recognition stage and this is implemented by Viterbi decoding with modified transition matrices for both cases (in GMM the transition matrix is created from scratch while in HMM the changes are merged into η). Such a strategy allows the hidden dynamic model to be used in phone recognition directly *without* resorting to an N-best list provided by HMM.

6.4 Discriminative Deep Neural Networks Aided by Generative Pre-training

After providing detailed reviews of a range of generative deep-structured dynamic models of speech, we now turn to their discriminative counterpart. Recall that the generative model expressed by (6.4) and (6.5) have deep structure, with causal relations from the top discrete linguistic symbols s through to hidden dynamic vectors and then to the bottom observed vectors. The reverse direction, from bottom to top, is referred to as the inference step, which is required to perform learning (i.e., training) and for decoding in speech recognition whose goal is to estimate the linguistic symbol sequences. Now we discuss the discriminative version of the deep-structured models, where the direct information flow is opposite: bottom up rather than top down. That is, the observed acoustic variables are used to directly compute the intermediate representations, and then to compute the estimate of linguistic labels. It turns out that the deep neural network (DNN) is an excellent candidate for this type of model, as (non-recurrent) neural networks are known to lack the modeling power for explicit speech dynamics.

Historically, the DNN had been very difficult to learn before 2006 [11, 80]. The difficulty was alleviated around 2006 with the work of [52, 53], where a generative pre-training procedure was developed and reported. In this section, we will review this advance and the more recent impact by the DNN on speech recognition research and deployment. We will then analyze the weaknesses of the DNN-based methods, especially those in modeling speech dynamics. This analysis paves a natural path to the recurrent versions of the DNN as well as their connections to and the differences between the generative deep-structured dynamic models of speech reviewed in the preceding two sections.

6.4.1 Restricted Boltzmann Machines

The generative pre-training procedure first reported in [52, 53] starts with the restricted Boltzmann machine (RBM), which is a special type of Markov random field that has one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible or observable units.

In an RBM, the joint distribution $p(\mathbf{v}, \mathbf{h}; \theta)$ over the visible units $\mathbf{v}$ and hidden units $\mathbf{h}$, given the model parameters θ, is defined in terms of an energy function $\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta)$ of

$$p(\mathbf{v}, \mathbf{h}; \theta) = \frac{\exp(-\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta))}{Z}, \tag{6.40}$$

where $Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta))$ is a normalization factor, and the marginal distribution that the model assigns to a visible vector $\mathbf{v}$ (we don't care about $\mathbf{h}$ since it is hidden) is

$$p(\mathbf{v}; \theta) = \frac{\sum_{\mathbf{h}} \exp(-\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta))}{Z}. \tag{6.41}$$

For a Bernoulli (visible)-Bernoulli (hidden) RBM, the energy function is defined as

$$\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij} v_i h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j, \tag{6.42}$$

where w_{ij} represents the symmetric interaction term between the visible unit v_i and the hidden unit h_j, b_i and a_j are the bias terms, and I and J are the numbers of visible and hidden units. The conditional distributions (for Bernoulli stochastic variables, i.e. binary data) can be efficiently calculated as

$$p(h_j = 1 | \mathbf{v}; \theta) = \sigma\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right), \tag{6.43}$$

$$p(v_i = 1 | \mathbf{h}; \theta) = \sigma\left(\sum_{j=1}^{J} w_{ij} h_j + b_i\right) \tag{6.44}$$

where $\sigma(x) = 1/(1 + \exp(x))$.

Similarly, for a Gaussian (visible)-Bernoulli (hidden) RBM, the energy is

$$\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij} v_i h_j - \frac{1}{2} \sum_{i=1}^{I} (v_i - b_i)^2 - \sum_{j=1}^{J} a_j h_j, \tag{6.45}$$

The corresponding conditional distributions (for Bernoulli or binary $\mathbf{h}$ and Gaussian or continuous-valued $\mathbf{v}$) become

$$p\left(h_j = 1|\mathbf{v};\theta\right) = \sigma\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right), \tag{6.46}$$

$$p\left(v_i|\mathbf{h};\theta\right) = \mathcal{N}\left(\sum_{j=1}^{J} w_{ij} h_j + b_i,\ 1\right), \tag{6.47}$$

where v_i takes real values and follows a Gaussian distribution with mean $\sum_{j=1}^{J} w_{ij} h_j + b_i$ and variance one. Gaussian-Bernoulli RBMs can be used to convert real-valued stochastic variables to binary stochastic variables, which can then be further processed using the Bernoulli-Bernoulli RBMs.

Taking the gradient of the log likelihood $\log p\left(\mathbf{v};\theta\right)$ we can derive the update rule for the RBM weights as

$$\triangle w_{ij} \propto E_{data}\left(v_i h_j\right) - E_{model}\left(v_i h_j\right), \tag{6.48}$$

where $E_{data}\left(v_i h_j\right)$ is the expectation observed in the training set under the distribution defined by the given observations, $p(h|\mathrm{v};\theta)$, and $E_{model}\left(v_i h_j\right)$ is that same expectation under the distribution defined by the model, $p\left(\mathbf{v},\mathbf{h};\theta\right)$. Calculation of $E_{data}\left(v_i h_j\right)$ is facilitated by using $p(h_j = 1|\mathbf{v};\theta)$ to weight samples $v_i h_j$ given observations $\mathbf{v}$. Unfortunately, $E_{model}\left(v_i h_j\right)$ is intractable to compute so the contrastive divergence (CD) approximation to the gradient is used where $E_{model}\left(v_i h_j\right)$ is replaced by running the Gibbs sampler initialized at the data for one full step. The steps in approximating $E_{model}\left(v_i h_j\right)$ are as follows:

1. Initialize $\mathbf{v_0}$ by the data
2. Sample $\mathbf{h_0} \sim p\left(\mathbf{h}|\mathbf{v_0}\right)$
3. Sample $\mathbf{v_1} \sim p\left(\mathbf{v}|\mathbf{h_0}\right)$
4. Sample $\mathbf{h_1} \sim p\left(\mathbf{h}|\mathbf{v_1}\right)$

Then the $(\mathbf{v_1},\mathbf{h_1})$ is a sample from the model, acting as a very rough estimate of $E_{model}\left(v_i h_j\right)$. Use of $(\mathbf{v_1},\mathbf{h_1})$ to approximate $E_{model}\left(v_i h_j\right)$ gives rise to the algorithm of CD-1. The sampling process is pictorially depicted in Fig. 6.2 where $< v_i h_j >^k \equiv (\mathbf{v}_k,\mathbf{h}_k)$.

Careful training of RBMs is essential to the success of applying the RBM and related deep learning techniques to solve practical problems. See the technical report [50] for a very useful practical guide for training RBMs.

The RBM discussed above is a generative model, which characterizes the input data distribution using hidden variables and there is no label information involved. However, when the label information is available, it can be used together with the data to form the joint "data" set. Then the same CD learning can be applied to

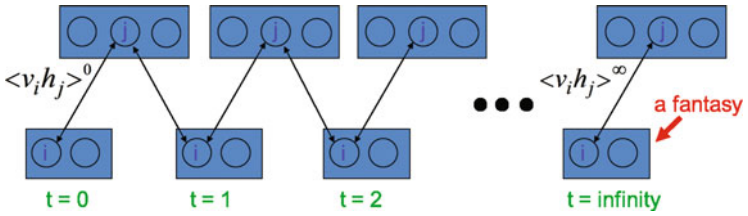

Fig. 6.2 A pictorial view of sampling from an RBM during RBM learning (courtesy of Geoff Hinton)

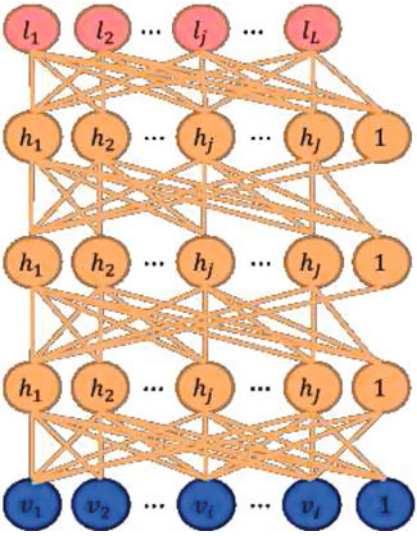

Fig. 6.3 An illustration of the DBN/DNN architecture

optimize the approximate "generative" objective function related to data likelihood. Further, and more interestingly, a "discriminative" objective function can be defined in terms of the conditional likelihood of labels. This discriminative RBM can be used to "fine tune" an RBM for classification tasks [60].

6.4.2 Stacking Up RBMs to Form a DBN

Stacking a number of the RBMs learned layer by layer from bottom up gives rise to a deep belief network (DBN), an example of which is shown in Fig. 6.3. The stacking procedure is as follows. After learning a Gaussian-Bernoulli RBM (for applications with continuous features such as speech) or Bernoulli-Bernoulli RBM (for applications with nominal or binary features such as a black-white image or coded text), we treat the activation probabilities of its hidden units as the data for

training the Bernoulli-Bernoulli RBM one layer up. The activation probabilities of the second-layer Bernoulli-Bernoulli RBM are then used as the visible data input for the third-layer Bernoulli-Bernoulli RBM, and so on. Mathematically for a DBN with M layers we can model the joint distribution between the observations $\mathbf{v}$ and the L hidden layers $\{\mathbf{h}^k : k = 1, 2, \ldots M\}$ as follows

$$p\left(\mathbf{v}, \mathbf{h}^1, \ldots, \mathbf{h}^M\right) = p\left(\mathbf{v} \mid \mathbf{h}^1\right) \left(\prod_{k=1}^{M-2} p(\mathbf{h}^k | \mathbf{h}^{k+1}) \right) p(\mathbf{h}^{M-1}, \mathbf{h}^M) \tag{6.49}$$

This allows us to derive relevant distributions, e.g. the posterior distribution $p(\mathbf{h}^M|\mathbf{v})$. Some theoretical justification of this efficient layer-by-layer greedy learning strategy is given in [52], where it is shown that the *stacking* procedure above improves a variational lower bound on the likelihood of the training data under the composite model. That is, the greedy procedure above achieves approximate maximum likelihood learning. Note that this learning procedure is unsupervised and requires no class label. Each hidden layer can be considered a nonlinear feature detector of the previous hidden layer outputs and each layer adds progressively more complex statistical structure by the data with the top layer representing the highest level desired structure.

When applied to classification tasks, the generative pre-training can be followed by or combined with other, typically discriminative, learning procedures that fine-tune all of the weights jointly to improve the performance of the network. This discriminative fine-tuning is performed by adding a final layer of variables that represent the desired outputs or labels provided in the training data. Then, the back-propagation algorithm can be used to adjust or fine-tune the network weights in the same way as for the standard feed-forward neural network. When used in this way we refer to this as the deterministic neural network or DNN. What goes to the top, label layer of this DNN depends on the application. For speech recognition applications, the top layer, denoted by $\mathbf{h}^M = \{l_1, l_2, \ldots, l_j, \ldots, l_L\}$, in Fig. 6.3, can represent either syllables, phones, sub-phones, phone states, or other speech units used in the HMM-based speech recognition system.

The generative pre-training described above has produced better phone and speech recognition results than random initialization on a wide variety of tasks. Further research has also shown the effectiveness of other pre-training strategies. As an example, greedy layer-by-layer training may be carried out with an additional discriminative term to the generative cost function at each level. And without generative pre-training, purely discriminative training of DNNs from random initial weights using the traditional stochastic gradient decent method has been shown to work very well when the scales of the initial weights are set carefully and the mini-batch sizes, which trade off noisy gradients with convergence speed, used in stochastic gradient decent are adapted prudently (e.g., with an increasing size over training epochs). Also, randomization order in creating mini-batches needs to be judiciously determined. Importantly, it was found effective to learn a DNN by starting with a shallow neural net with a single hidden layer. Once this has been

trained discriminatively (using early stops to avoid overfitting), a second hidden layer is inserted between the first hidden layer and the labeled softmax output units and the expanded deeper network is again trained discriminatively. This can be continued until the desired number of hidden layers is reached, after which a full backpropagation "fine tuning" is applied. This discriminative "pre-training" procedure is found to work well in practice (e.g., [94, 107]).

Despite the great success in using DNNs for large vocabulary speech recognition, training is still quite slow due to the large number of parameters and the required large data set sizes. Part of current research has now begun to focus on optimization techniques to improve the training regime for DNNs [93] specifically and for speech and language processing as a whole [104].

6.4.3 Interfacing the DNN with an HMM to Incorporate Sequential Dynamics

The DNN discussed above is a static classifier with input vectors having a fixed dimensionality. However, many practical pattern recognition and information processing problems, including speech recognition, machine translation, natural language understanding, video processing and bio-information processing, require sequence recognition. In sequence recognition, sometimes called classification with structured input/output, the dimensionality of both inputs and outputs are variable.

The HMM, based on dynamic programming operations, is a convenient tool to help port the strength of a static classifier to handle dynamic or sequential patterns. Thus, it is natural to combine the DNN and HMM to bridge the gap between static and sequence pattern recognition. A popular architecture to fulfil this is shown in Fig. 6.4. This architecture has been successfully used in speech recognition experiments from small to mid and to large scales, as reported in [15, 16, 51, 59, 64, 77, 79, 93, 94, 108–110]. The excellent recognition accuracy obtained by the DNN-HMM and its scalability from small to large tasks have resulted in wide industry adoption of this architecture and a huge surge of research efforts. This is so despite the recognition of the weaknesses of modeling realistic speech dynamics via the HMM and via the windowed speech frames as inputs to the DNN.

It is important to note that the unique elasticity of the temporal dynamic of speech as elaborated in [39, 40] would require temporally-correlated models more powerful than the HMM for the ultimate success of speech recognition. Integrating such dynamic models having realistic co-articulatory properties with the DNN and possibly other deep learning models to form the coherent dynamic deep architecture is a challenging new research. Adding recurrent connections over the hidden neurons gives one reasonable way of incorporating speech dynamics into the model, at least more principled than using a long window of frames in the DNN-HMM architecture. In the next section we turn to a review and analysis of the recurrent neural network (RNN) before providing connections to the generative deep-structured dynamic speech models reviewed earlier.

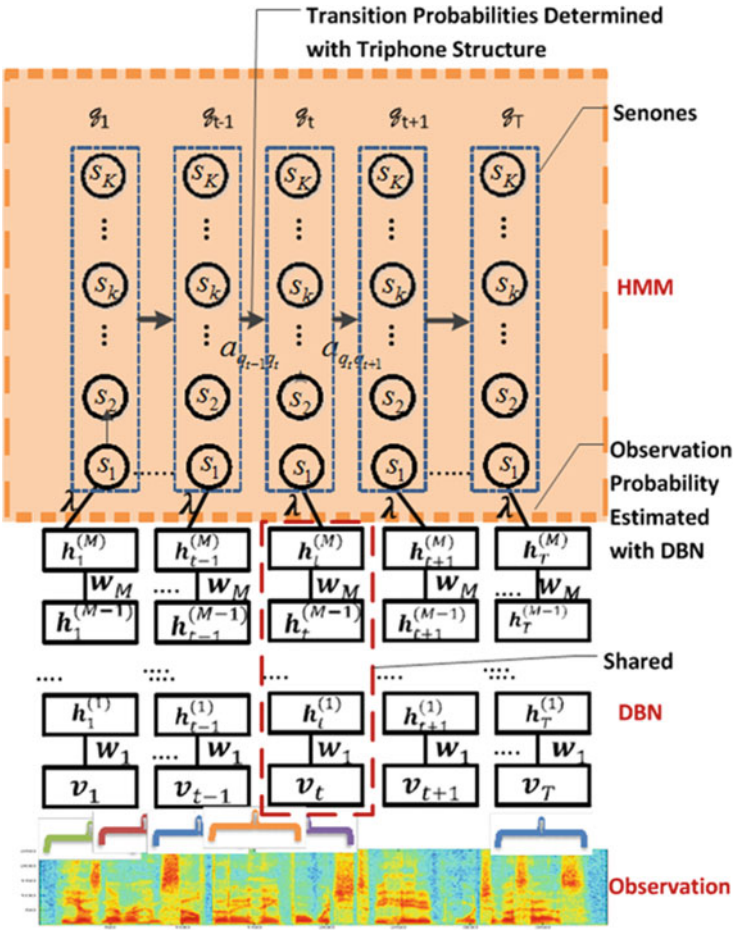

Fig. 6.4 Interface between DBN/DNN and HMM to form a DBN-HMM or DNN-HMM

6.5 Recurrent Neural Networks for Discriminative Modeling of Speech Dynamics

The use of RNNs or related neural predictive models for speech recognition dates back to early 1990s (e.g., [27, 90]), with relatively low accuracy and whose results could not be reproduced by other groups until recently. Since deep learning became popular in recent years, much more research has been devoted to the RNN (e.g., [48, 69–71, 73–76, 84, 98–100, 103] and its stacked versions, also called deep RNNs [49]. Most work on RNNs made use of the method of Back Propagation Through Time (BPTT) to train the RNNs, and empirical tricks need to be exploited (e.g., truncate gradients when they become too large [74]) in order to make

the training effective. It is not until recently that careful analysis was made to fully understand the source of difficulties in learning RNNs and somewhat more principled, but still rather heuristic, solutions were developed. For example, in [7, 84], a heuristic strategy of gradient norm clipping was proposed to deal with the gradient exploding problem during BPTT training. There are other solutions offered to improve the learning method for the RNN (e.g., [28, 56]).

6.5.1 RNNs Expressed in the State-Space Formalism

Let us formulate the RNN in terms of the nonlinear state space model commonly used in signal processing. We will compare it with the same state space formulation of nonlinear dynamic systems used as generative models for speech acoustics. The contrast between the discriminative RNN and the use of the same mathematical model in the generative mode allows us to shed light onto why one approach works better than another and how a combination of the two is desirable.

As shown in Fig. 6.5 given an input sequence $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T)$, the RNN computes the noise free hidden state dynamic vector sequence $\mathbf{H} = (\mathbf{h}_1, \cdots, \mathbf{h}_t, \cdots, \mathbf{h}_T)$ by iterating the following from $t = 1$ to T:

$$\mathbf{h}_t = f(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1}) = f(\mathbf{u}_t) \tag{6.50}$$

$$\mathbf{y}_t = g(\mathbf{W}_{hy}\mathbf{h}_t) = g(\mathbf{v}_t) \tag{6.51}$$

where $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_t, \cdots, \mathbf{y}_T)$ is the "target label" output sequence, which is the "observation" sequence in the standard state-space formulation.

The desired target signal in the above state-space model is the predicted "label" or target vector, $\mathbf{l}_t$, a vector of one-hot coded class labels. Define the error function

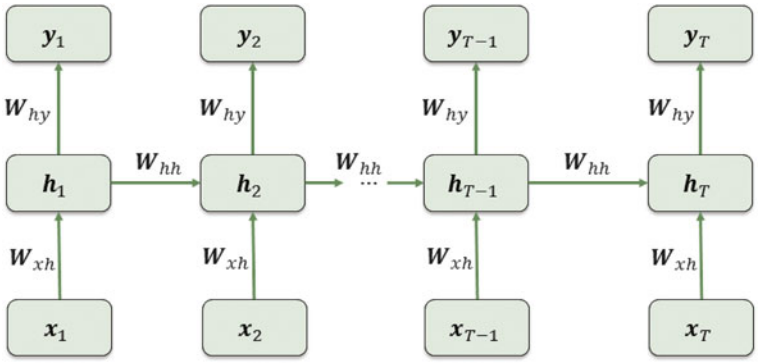

Fig. 6.5 Information flow in the standard recurrent neural network from observation variables to the target labels as output variables via the hidden-state vectors

as the sum of squared differences between $\mathbf{y}_t$ and $\mathbf{l}_t$ over time, or the cross entropy between them. Then BPTT unfolds the RNN over time in computing the gradients with respect to $\mathbf{W}_{hy}, \mathbf{W}_{xh}$ and $\mathbf{W}_{hh}$, and stochastic gradient descent is applied to update these weight matrices.

6.5.2 The BPTT Learning Algorithm

The BPTT [10,56] is an extension of the classic feedforward backpropagation where the stacked hidden layers for the same training epoch, t, are replaced by unfolding the recurrent neural network in time and stacking T single hidden layers across time, $t = 1, 2, \ldots, T$. Referring to Fig. 6.5 and (6.50), (6.51) let us assume a recurrent neural network with K inputs, N internal hidden units, and L outputs, and define the following variables at time layer t:

- $\mathbf{x}_t$ is the $K \times 1$ vector of inputs, $\mathbf{h}_t$ is the $N \times 1$ vector of hidden unit outputs, $\mathbf{y}_t$ is the $L \times 1$ vector of outputs, and $\mathbf{l}_t$ is the $L \times 1$ vector of training output targets, where the jth vector element, e.g., $h_t(j)$ is the jth hidden unit for $j = 1, 2, \ldots, N$;
- $\mathbf{W}_{hy}$ is the $L \times N$ matrix of weights connecting the N hidden units to the L outputs, $\mathbf{W}_{xh}$ is the $N \times K$ matrix of weights connecting the K inputs to the N hidden units, and $\mathbf{W}_{hh}$ is the $N \times N$ matrix of weights connecting the N hidden units from layer $t-1$ to layer t, where the (i, j)th matrix element, e.g., $w_{hy}(i, j)$ is the weight connecting the jth hidden unit to the ith output unit for $i = 1, 2, \ldots, L$ and $j = 1, 2, \ldots, N$;
- $\mathbf{u}_t = \mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1}$ is the $N \times 1$ vector of hidden unit input potentials, $\mathbf{v}_t = \mathbf{W}_{hy}\mathbf{h}_t$ is the $L \times 1$ vector of output unit input potentials, from which we have $\mathbf{h}_t = f(\mathbf{u}_t)$ and $\mathbf{y}_t = g(\mathbf{v}_t)$; where
- $f(\mathbf{u}_t)$ is the hidden layer activation function ($f'(\mathbf{u}_t)$ is its derivative), and $g(\mathbf{v}_t)$ is the output layer activation function ($g'(\mathbf{v}_t)$ is its derivative).

Similar to classic backpropagation we begin by defining the summed squared error between the actual output, $\mathbf{y}_t$, and the target vector, $\mathbf{l}_t$, averaged across all time epochs:

$$E = c \sum_{t=1}^{T} \|\mathbf{l}_t - \mathbf{y}_t\|^2 = c \sum_{t=1}^{T} \sum_{j=1}^{L} (l_t(j) - y_t(j))^2 \tag{6.52}$$

where c is a conveniently chosen scale factor and seek to minimise this error w.r.t to the weights using a gradient descent. For a specific weight, w, the update rule for gradient descent is:

$$w^{new} = w - \gamma \frac{\partial E}{\partial w} \tag{6.53}$$

To do this we define the so-called error propagation term which is the error gradient w.r.t to the unit input potential:

$$\delta_t^y(j) = -\frac{\partial E}{\partial v_t(j)}, \quad \delta_t^h(j) = -\frac{\partial E}{\partial u_t(j)} \tag{6.54}$$

choose $c = 0.5$ and then use the chain rule (keeping track of the dependencies) as follows:

1. For $t = 1, 2, \ldots, T$ compute the input potentials $(\mathbf{u}_t, \mathbf{v}_t)$ and activation outputs $(\mathbf{h}_t, \mathbf{y}_t)$ given the current RNN weights and input $\mathbf{x}_t$ (the forward pass).
2. At time layer $t = T$ calculate the error propagation term (where $\odot$ is the component-wise multiplication operator):

$$\delta_T^y(j) = -\frac{\partial E}{\partial y_T(j)}\frac{\partial y_T(j)}{\partial v_T(j)} = (l_T(j) - y_T(j))g'(v_T(j)) \quad \text{for } j = 1, 2, \ldots, L$$

$$\boldsymbol{\delta}_T^y = (\mathbf{l}_T - \mathbf{y}_T) \odot g'(\mathbf{v}_T) \tag{6.55}$$

at the output units and

$$\delta_T^h(j) = -\left(\sum_{i=1}^{L}\frac{\partial E}{\partial v_T(i)}\frac{\partial v_T(i)}{\partial h_T(j)}\frac{\partial h_T(j)}{\partial u_T(j)}\right) = \sum_{i=1}^{L}\delta_T^y(i)w_{hy}(i,j)f'(u_T(j)) \quad \text{for } j = 1, 2, \ldots, N$$

$$\boldsymbol{\delta}_T^h = \mathbf{W}_{hy}^T\boldsymbol{\delta}_T^y \odot f'(\mathbf{u}_T) \tag{6.56}$$

for the internal units (where $\boldsymbol{\delta}_T^y$ is propagated back from the output layer T).

3. At the earlier layers, $t = T-1, T-2, \ldots, 1$, calculate the error propagation term:

$$\delta_t^y(j) = (l_t(j) - y_t(j))g'(v_t(j)) \quad \text{for } j = 1, 2, \ldots, L$$

$$\boldsymbol{\delta}_t^y = (\mathbf{l}_t - \mathbf{y}_t) \odot g'(\mathbf{v}_t) \tag{6.57}$$

for the output units and

$$\delta_t^h(j) = -\left[\sum_{i=1}^{N}\frac{\partial E}{\partial u_{t+1}(i)}\frac{\partial u_{t+1}(i)}{\partial h_t(j)} + \sum_{i=1}^{L}\frac{\partial E}{\partial v_t(i)}\frac{\partial v_t(i)}{\partial h_t(j)}\right]\frac{\partial h_t(j)}{\partial u_t(j)}$$

$$= \left[\sum_{i=1}^{N}\delta_{t+1}^h(i)w_{hh}(i,j) + \sum_{i=1}^{L}\delta_t^y(i)w_{hy}(i,j)\right]f'(u_t(j)) \quad \text{for } j = 1, 2, \ldots, N$$

$$\boldsymbol{\delta}_t^h = \left[\mathbf{W}_{hh}^T\boldsymbol{\delta}_{t+1}^h + \mathbf{W}_{hy}^T\boldsymbol{\delta}_t^y\right] \odot f'(\mathbf{u}_t) \tag{6.58}$$

for the internal units (where $\boldsymbol{\delta}_t^y$ is propagated back from the output layer t, and $\boldsymbol{\delta}_{t+1}^h$ is propagated back from hidden layer $t+1$).

Then we adjust the weights as follows:

1. For the jth hidden to ith output layer weights at layer t:

$$w_{hy}^{new}(i,j) = w_{hy}(i,j) - \gamma \sum_{t=1}^{T} \frac{\partial E}{\partial v_t(i)} \frac{\partial v_t(i)}{\partial w_{hy}(i,j)} = w_{hy}(i,j) - \gamma \sum_{t=1}^{T} \delta_t^y(i) h_t(j)$$

$$\mathbf{W}_{hy}^{new} = \mathbf{W}_{hy} + \gamma \sum_{t=1}^{T} \boldsymbol{\delta}_y^t \mathbf{h}_t^T \tag{6.59}$$

2. For the jth input to the ith hidden layer weights at layer t:

$$w_{xh}^{new}(i,j) = w_{xh}(i,j) - \gamma \sum_{t=1}^{T} \frac{\partial E}{\partial u_t(i)} \frac{\partial u_t(i)}{\partial w_{xh}(i,j)} = w_{xh}(i,j) - \gamma \sum_{t=1}^{T} \delta_t^h(i) x_t(j)$$

$$\mathbf{W}_{xh}^{new} = \mathbf{W}_{xh} + \gamma \sum_{t=1}^{T} \boldsymbol{\delta}_h^t \mathbf{x}_t^T \tag{6.60}$$

3. For the jth hidden at layer $t+1$ to the ith hidden at layer t weights:

$$w_{hh}^{new}(i,j) = w_{hh}(i,j) - \gamma \sum_{t=1}^{T} \frac{\partial E}{\partial u_t(i)} \frac{\partial u_t(i)}{\partial w_{hh}(i,j)} = w_{xhh}(i,j) - \gamma \sum_{t=1}^{T} \delta_t^h(i) h_{t-1}(j)$$

$$\mathbf{W}_{hh}^{new} = \mathbf{W}_{hh} + \gamma \sum_{t=1}^{T} \boldsymbol{\delta}_h^t \mathbf{h}_{t-1}^T \tag{6.61}$$

where γ is the learning rate.

One drawback of the BPTT is that the entire time series is needed to perform one update of the weights, thereby making BPTT a "batch" adaptation algorithm. It is possible to consider an online adaptation if one truncates the past history to no more than the last p time epochs, creating the BPTT(p) or p-BPTT variant.

The computational complexity of the BPTT is given as $O(M^2)$ per time step where $M = LN + NK + N^2$ is the number of internal units. As with classic feedforward backpropagation slow convergence can be expected with several thousand epochs needed. However unlike feedforward backpropagation the BPTT is not guaranteed to converge to a local minimum and it is far from trivial to achieve good results without much experimentation and tuning. This is mainly due to the problem of exploding and vanishing gradients as described in [84].

6.5.3 The EKF Learning Algorithm

In Sect. 6.2.2 the extended Kalman filter (EKF) was used to provide estimates of the hidden state variable in the non-linear state-space system described by (6.4) and (6.5). By reformulating the state-space system such that the hidden state variable are the RNN weights and the system observations are the target vectors we can use the EKF as a learning algorithm for the RNN. First popularised in the hallmark work of [87] we proceed by restacking the $L \times N$ $\mathbf{W}_{hy}$, $N \times K$ $\mathbf{W}_{xh}$, and $N \times N$ $\mathbf{W}_{hh}$ RNN weights into a single, state vector $\mathbf{w}$ of size $LN + NK + N^2$. Then we form the following state-space system:

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \mathbf{q}(n) \\ \mathbf{l}(n) &= \mathbf{h}_n(\mathbf{w}(n), \mathbf{x}_{1:n}) \end{aligned} \tag{6.62}$$

where $\mathbf{l}(n) \equiv \mathbf{l}_n$ is the target vector and the desired "observation" from the system at time n, $\mathbf{q}(n)$ is the external input to the system considered as an uncorrelated Gaussian white noise process, $\mathbf{w}(n)$ are the RNN weights at time n, and $\mathbf{y}_n \equiv \mathbf{h}_n(\hat{\mathbf{w}}(n), \mathbf{x}_{1:n})$ is the time-dependent RNN output observation function at time-step n derived from the current RNN weight estimates $\hat{\mathbf{w}}(n)$ and the input vector sequence $\mathbf{x}_{1:n} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$. The EKF recursion applied to this system will estimate the unknown hidden state $\mathbf{w}(n)$, given the "observations" $\mathbf{l}(n)$, by attempting to minimise the innovation error $\boldsymbol{\xi}(n) = (\mathbf{l}(n) - \mathbf{h}_n(\hat{\mathbf{w}}(n), \mathbf{x}_{1:n})) = (\mathbf{l}_n - \mathbf{y}_n)$ in the minimum mean square error (MMSE) sense equivalent to the minimisation of the BPTT squared error of (6.52). The EKF recursion for this system simplifies to:

$$\begin{aligned} \mathbf{K}(n) &= \mathbf{P}(n)\mathbf{H}(n)[\mathbf{H}(n)^T\mathbf{P}(n)\mathbf{H}(n)]^{-1} \\ \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) + \mathbf{K}(n)\boldsymbol{\xi}(n) \\ \mathbf{P}(n+1) &= \mathbf{P}(n) - \mathbf{K}(n)\mathbf{H}(n)^T\mathbf{P}(n) + \mathbf{Q}(n) \end{aligned} \tag{6.63}$$

where $\mathbf{K}(n)$ is the Kalman gain, $\mathbf{P}(n) = E[(\mathbf{w}(n) - \hat{\mathbf{w}}(n))(\mathbf{w}(n) - \hat{\mathbf{w}}(n))^T$ is the state error covariance and $\mathbf{H}(n) = \left.\frac{\partial \mathbf{h}_n(\mathbf{w}(n)), \mathbf{x}_{1:n})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\hat{\mathbf{w}}(n)}$ is the Jacobian of partial derivatives of the RNN output with respect to the weights. The EKF recursion requires the initial estimates $\hat{\mathbf{w}}(0)$ and $\mathbf{P}(0)$ and a model for the process noise $\mathbf{Q}(n)$. Typically $\hat{\mathbf{w}}(0)$ is generated randomly, $\mathbf{P}(0)$ is set to a diagonal matrix with a large diagonal component and $\mathbf{Q}(n)$ is a diagonal matrix with small diagonal variance terms.

Although the EKF recursion is a very elegant approach exploiting the theory of optimum Kalman filters, the Jacobian linearisation of the non-linear h_n only guarantees convergence to a local minimum. Furthermore the calculation of the Jacobian at each iteration time step requires either direct calculation of the gradients, which is computationally expensive, or the use of an offline run of the BPTT for the

gradients by backpropagation. The BPTT-EKF executes a reformulated BPTT(p) over the input data sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, where the RNN weights $\mathbf{w} = \hat{\mathbf{w}}(n)$, to calculate the gradient $\frac{\partial \mathbf{y}_n}{\partial \mathbf{w}}$ for the $\mathbf{H}(n)$. This is followed by one iteration of the EKF recursion to calculate $\hat{\mathbf{w}}(n+1)$ and so on. The BPTT-EKF exhibits an order $O(LM^2)$ computational complexity per time-step where $M = LN + NK + N^2$ is the number of internal units and has been shown to exhibit superior convergence over BPTT and can be considered one of the classic state of the art approaches to RNN training.

6.6 Comparing Two Types of Dynamic Models

We are now in a position to discuss similarities of and differences between the two types of deep and dynamic models: (1) the generative deep-structured dynamic model, which we reviewed in Sect. 6.2, and (2) the discriminative RNN, which we reviewed in Sect. 6.5. The "deepness" of the models is expressed in terms of the time steps. Several key aspects are compared below, one in each subsection.

6.6.1 Top-Down Versus Bottom-Up

Top-down modeling here refers to the hierarchical way in which the speech data are modeled by the generative hidden dynamics. The modeling process starts with specification of the linguistic label sequence at the top level. Then the label sequence generates the hidden dynamic vector sequence, which in turn generates the acoustic observation sequence at the bottom level in the hierarchy. This way of modeling can be viewed as fitting the observation data. On the other hand, in bottom-up modeling based on the RNN, the information flow starts at the bottom level of acoustic observation, which activates the hidden layer or vector dynamics in the RNN. Then the output layer of the RNN computes the linguistic label or target sequence at the top level of the hierarchy. Since the top layer determines the speech-class distinction, the bottom-up modeling approach can also be called discriminative learning. We elaborate on the top-down versus bottom-up comparisons below.

6.6.1.1 Top-Down Generative Hidden Dynamic Modeling

To facilitate the comparison, we use a general form of the generative hidden dynamic model following the discussion in Sect. III.A of [33] with slight modification, and note that speech recognition researchers have used many variants of this form to build speech recognizers in the past; see a survey in Sects. III.D and III.E of [33] and the review in Sect. 6.2. In the discussion provided in this section, the general form of the state and observation equations in the generative hidden dynamic model takes the form of

$$\mathbf{h}_t = G(\mathbf{h}_{t-1}; \mathbf{W}_{l_t}, \mathbf{\Lambda}_{l_t}) + \text{StateNoise} \tag{6.64}$$

$$\mathbf{x}_t = H(\mathbf{h}_t, \mathbf{\Omega}_{l_t}) + \text{ObsNoise} \tag{6.65}$$

Here, $\mathbf{W}_{l_t}$ is the system matrix that shapes the (articulatory-like) state dynamics, and $\mathbf{\Lambda}_{l_t}$ serves as the "input" driving force to the state dynamics. Both of them are dependent on the label l_t at time t with segmental properties, hence the model is also called a (segmental) switching dynamic system. The system matrix is analogous to $\mathbf{W}_{hh}$ in the RNN. $\mathbf{\Omega}_{l_t}$ is the parameter set that governs the nonlinear mapping from the hidden (articulatory-like) states in speech production to acoustic features of speech. In one implementation, $\mathbf{\Omega}_{l_t}$ took the form of shallow MLP weights [35, 86, 101]. In another implementation, $\mathbf{\Omega}_{l_t}$ took the form of a set of matrices in a mixture of linear experts [67].

The state equation in various previous implementations of the hidden dynamic models of speech does not take nonlinear forms. Rather, the following linear form was used (e.g., [35], as we discussed in Sect. 6.2.2):

$$\mathbf{h}_t = \mathbf{W}_{hh}(l_t)\mathbf{h}_{t-1} + [\mathbf{I} - \mathbf{W}_{hh}(l_t)]\mathbf{t}_{l_t} + \text{StateNoise} \tag{6.66}$$

which exhibits the target-directed property for the articulatory-like dynamics. Here, the parameters $\mathbf{W}_{hh}$ are a function of the (phonetic) label l_t at a particular time t, and $\mathbf{t}_{l_t}$ is a mapping from the symbolic quantity l_t of a linguistic unit to the continuous-valued "target" vector with the segmental property. To make the following comparisons easy, let's keep the nonlinear form and remove both the state and observation noise, yielding the state-space generative model of

$$\mathbf{h}_t = G(\mathbf{h}_{t-1}; \mathbf{W}_{l_t}, \mathbf{t}_{l_t}) \tag{6.67}$$

$$\mathbf{x}_t = H(\mathbf{h}_t, \mathbf{\Omega}_{l_t}) \tag{6.68}$$

6.6.1.2 Bottom-Up Discriminative Recurrent Neural Networks and the "Generative" Counterpart

Let us rewrite (6.50) and (6.51) into a more general form:

$$\mathbf{h}_t = F(\mathbf{h}_{t-1}, \mathbf{x}_t; \mathbf{W}_{hh}, \mathbf{W}_{xh}) \tag{6.69}$$

$$\mathbf{y}_t = K(\mathbf{h}_t; \mathbf{W}_{hy}). \tag{6.70}$$

where information flow goes from observation data $\mathbf{x}_t$ to hidden vectors $\mathbf{h}_t$ and then to the predicted target label vectors $\mathbf{y}_t$ in the bottom-up direction.

Compared with (6.67) and (6.68), which describe the information flow from the top-level label-indexed phonetic "target" vector $\mathbf{t}_{l_t}$ to hidden vectors $\mathbf{h}_t$ and then to observation data $\mathbf{x}_t$, we clearly see opposite information flows.

In order to examine other differences between the two types of models in addition to the top-down versus bottom-up difference, we keep the same mathematical description of the RNN but swap the variables of input $\mathbf{x}_t$ and output $\mathbf{y}_t$ in (6.69) and (6.70). This yields

$$\mathbf{h}_t = F_1(\mathbf{h}_{t-1}, \mathbf{y}_t; \mathbf{W}_{hh}, \mathbf{W}_{yh}) \tag{6.71}$$

$$\mathbf{x}_t = K_1(\mathbf{h}_t; \mathbf{W}_{hx}). \tag{6.72}$$

or more specifically

$$\mathbf{h}_t = f_1(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{yh}\mathbf{y}_t) \tag{6.73}$$

$$\mathbf{x}_t = g_1(\mathbf{W}_{hx}\mathbf{h}_t) \tag{6.74}$$

The "generative" version of the RNN can be illustrated by Fig. 6.6, which is the same as the normal "discriminative" version of the RNN shown in Fig. 6.5 except all arrows change their directions.

Given the "generative" form of the two types of deep, dynamic models, one (the hidden dynamic model) described by (6.67) and (6.68), and the other (the RNN) by (6.71) and (6.72), we discuss below the contrast between them with respect to the different nature of the hidden-space representations (while keeping the same generative form of the models). We will also discuss below other aspects of the contrast between them including different ways of exploiting model parameters.

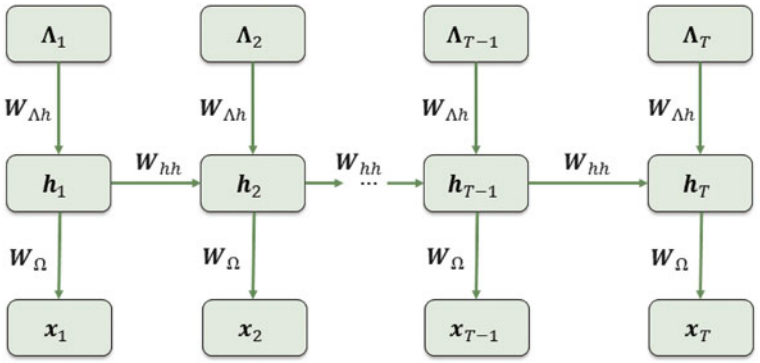

Fig. 6.6 Information flow in the same recurrent neural network of Fig. 6.5 except we swap the observation variables with the output variables without changing the mathematical form of the state-space model

6.6.2 Localist Versus Distributed Representations

Localist and distributed representations are important concepts in cognitive science as two distinct styles of data representation. In the localist representation, each neuron represents a single concept on a stand-alone basis. That is, localist units have their own meaning and interpretation, not so for the units in distributed representation. The latter pertains to an internal representation of concepts in such a way that they are modeled as being explained by the interactions of many hidden factors. A particular factor learned from configurations of other factors can often generalize well to new configurations, not so in localist representation.

Distributed representations, based on vectors consisting of many elements or units, naturally occur in a "connectionist" neural network, where a concept is represented by a pattern of activity across a number of many units and where at the same time a unit typically contributes to many concepts. One key advantage of such many-to-many correspondence is that they provide robustness in representing the internal structure of the data in terms of graceful degradation and damage resistance. Such robustness is enabled by redundant storage of information. Another advantage is that they facilitate automatic generalization of concepts and relations, thus enabling reasoning abilities. Further, distributed representation allows similar vectors to be associated with similar concepts and it allows efficient use of representational resources. These attractive properties of distributed representations, however, come with a set of weaknesses. These include non-obviousness in interpreting the representations, difficulties with representing hierarchical structure, and inconvenience in representing variable-length sequences. Distributed representations are also not directly suitable for input and output to a network and some translation with localist representations are needed.

On the other hand, local representation has advantages of explicitness and ease of use—the explicit representation of the components of a task is simple and the design of representational schemes for structured objects is easy. But the weaknesses are many, including inefficiency for large sets of objects, highly redundant use of connections, and undesirable growth of units in networks which represent complex structure.

All versions of the hidden dynamic models for deep speech structure [12, 19, 34, 37, 86, 101] adopt the "localist" representation of the symbolic linguistic units, and the RNN makes use of the distributed representation. This can be seen directly from (6.67) for the hidden dynamic model and from (6.71) for the RNN (in the "generative" version). In the former, symbolic linguistic units l_t as a function of time t are coded implicitly in a stand-alone fashion. The connection of symbolic linguistic units to continuous-valued vectors is made via a one-to-one mapping, denoted by $\mathbf{t}_{l_t}$ in (6.67), to the hidden dynamic's asymptotic "targets" denoted by vector $\mathbf{t}$. This type of mapping is common in phonetic-oriented phonology literature, and is called the "interface between phonology and phonetics" in a functional computational model of speech production in [19]. Further, the hidden dynamic model uses the linguistic labels represented in a localist manner to index separate

sets of time-varying parameters $\mathbf{W}_{l_t}$ and $\mathbf{\Omega}_{l_t}$, leading to "switching" dynamics which considerably complicates the decoding computation. This kind of parameter specification isolates the parameter interactions across different linguistic labels, gaining the advantage of explicit interpretation of the model but losing on direct discrimination across linguistic labels.

In contrast, in the state equation of the RNN model shown in (6.71), the symbolic linguistic units are directly represented as one-hot vectors of $\mathbf{y}_t$ as a function of time t. No mapping to separate continuous-valued "phonetic" vectors are needed. While the one-hot coding of $\mathbf{y}_t$ vectors is localist, the hidden state vector $\mathbf{h}$ provides a distributed representation and thus allows the model to store a lot of information about the past in a highly efficient manner. Importantly, there is no longer a notion of label-specific parameter sets of $\mathbf{W}_{l_t}$ and $\mathbf{\Omega}_{l_t}$ as in the hidden dynamic model. The weight parameters in the RNN are shared across all linguistic label classes. This enables direct discriminative learning for the RNN. In addition, the distributed representation used by the hidden layer of the RNN allows efficient and redundant storage of information, and has the capacity to automatically disentangle variation factors embedded in the data. However, as inherent in distributed representations discussed earlier, the RNN also carries with them the difficulty of interpreting the parameters and hidden states, and the difficulty of modeling structure.

6.6.3 *Latent Explanatory Variables Versus End-to-End Discriminative Learning*

An obvious strength of the localist representation as adopted by the hidden dynamic models for deep speech structure is that the model parameters and the latent (i.e. hidden) state variables are explainable and easy to diagnose. In fact, one main motivation of many of such models is that the knowledge of hierarchical structure in speech production in terms of articulatory and vocal tract resonance dynamics can be directly (but approximately with a clear sense of the degree of approximation) incorporated into the design of the models [12, 19, 20, 22, 31, 37, 66, 68, 83, 102, 106]. Practical benefits of using interpretable, localist representation of hidden state vectors include sensible ways of initializing the parameters to be learned (e.g., with extracted formants for initializing hidden variables composed of vocal tract resonances), and straightforward methods of diagnosing analyzing errors during model implementation. Since localist representations, unlike their distributed counterpart, do not superimpose patterns for signaling the presence of different linguistic labels, the hidden state variables not only are explanatory but also unambiguous. Further, the interpretable nature of the models allows complex causal and structured relationships to be built into them, free from the common difficulty associated with distributed representations. In fact, the hidden dynamic models have been constructed with many layers in the hierarchical hidden space, all with clear physical embodiment in speech production; e.g., Chap. 2 in [23]. However, the

complex structure makes it very difficult to do discriminative parameter learning. As a result, nearly all versions of hidden dynamic models have adopted maximum-likelihood learning or data fitting approaches. For example, the use of linear or nonlinear Kalman filtering (E step of the EM algorithm) for learning the parameters in the generative state-space models has been applied to only maximum likelihood estimates [95, 101].

In contrast, the learning algorithm of BPTT commonly used for end-to-end training of the RNN with distributed representations for the hidden states performs discriminative training by directly minimizing linguistic label prediction errors. It is straightforward to do so in the formulation of the learning objective because of each element in the hidden state vector contributes to all linguistic labels due to the very nature of the distributed representation. It is very unnatural and difficult to do so in the generative hidden dynamic model based on localist representations of the hidden states, where each state and the associated model parameters typically contribute to only one particular linguistic unit, which is used to index the set of model parameters.

6.6.4 Parsimonious Versus Massive Parameters

The final aspect of comparisons between the hidden dynamic model and the RNN concerns different ways to parameterize these two types of models. Due to the interpretable latent states in the hidden dynamic model as well as the parameters associated with them, speech knowledge can be used in the design of the model, leaving the size of free parameters to be relatively small. For example, when vocal tract resonance vectors are used to represent the hidden dynamics, a dimension of eight appears to be sufficient to capture the prominent dynamic properties responsible for the observed acoustic dynamics. Somewhat higher dimensionality is needed with the use of the hidden dynamic vectors associated with the articulators' configuration in speech production. The use of such parsimonious parameter sets, often called "small is good", is also facilitated by the localist representation of hidden state components and the related parameters that are connected or indexed to a specific linguistic unit. This contrasts with the distributed representation in the RNN where both the hidden state vector elements and the connecting weights are shared across all linguistic unit, thereby demanding many folds of more model parameters.

Although most of the current successful RNNs use fewer than one thousand hidden units and thus fewer than one million weight parameters in the recurrent matrix, we nevertheless call such parameterization "massive" for the following reasons. First, they are substantially larger than the set of parameters in any generative model of speech dynamics in the literature. Second, most of the current RNNs do not have special structure in their recurrent matrices and thus would not easily improve their discriminative power by blindly increasing the size of hidden layers. But with appropriate structure built into recurrent matrices, such as in the

"Long-Short-Term-Memory" version of the RNN, increasing the memory units and thus the parameters is expected to improve classification accuracy more markedly. Third, the neural network parameterization is not only easy to scale up, but it also gives regular computations using the same type of matrix multiplication regardless of the size of the matrices.

The ability to use speech-domain knowledge to construct the model with a parsimonious parameter set is both a blessing and a curse. Examples of such knowledge used in the past are the target-directed and smooth (i.e., non-oscillatory) hidden dynamics within each phone segment, an analytical relationship between the vocal tract resonance vector (both resonance frequencies and bandwidths), and both anticipatory and regressive types of coarticulation expressed in the latent space as a result of the hidden dynamics. With the right prediction of time-varying trajectories in the hidden space and then causally in the observed acoustic space, powerful constraints can be placed in the model formulation to reduce over-generation in the model space and to avoid unnecessarily large model capacity. On the other hand, the use of speech knowledge limits the growth of the model size as more data are made available in training. For example, when the dimensionality of the vocal tract resonance vectors goes beyond eight, many advantages of interpretable hidden vectors no longer hold. Since speech knowledge is necessarily incomplete, the constraints imposed on the model structure may be outweighed by the opportunity lost with increasingly large amounts of training data and by the incomplete knowledge.

In contrast, the RNN uses hardly any speech knowledge to constrain the model space due to the inherent difficulty of interpreting the ambiguous hidden state represented in a distributed manner. As such, the RNN in principle has the freedom to use massive parameters in keeping with the growing size of the training data. Lack of constraints may cause the model to over-generalize. This, together with the known difficulties of the various learning algorithms for the RNN as analyzed in [6] and reviewed in Sect. 6.5, has limited the progress of using RNNs in speech recognition for many years until recently. Some recent progress of RNNs applied to speech recognition involves various methods of introducing constraints either in the model construction or in the implementation of learning algorithms. For example, in the study reported in [49], the RNN's hidden state is designed with memory units, which, while constraining the variations of the recurrent hidden units and the associated weight parameters, still allow the massive model parameters to be used by simply increasing the size of the memory units. Separately, the RNN can also be constrained during the learning stage, where the size of the gradient computed by BPTT is limited by a threshold to avoid explosion as reported in [6, 75] or where the range of the permissible RNN parameters are constrained to be within what the "echo-state property" would allow [13, 25].

6.6.5 *Comparing Recognition Accuracy of the Two Types of Models*

Given the analysis on and comparisons presented so far in this section between the generative hidden dynamic model using localist representations and the discriminative RNN using distributed representations, we see both types of models have respective strengths and weaknesses. Here we compare the empirical performance of the two types of models in terms of speech recognition accuracy. For consistency reasons, we use the TIMIT phone recognition task for the comparison since no other common tasks have been used to assess both types of models in a consistent manner. It is important to point out that both types of the dynamic models are much more difficult to implement than other models in more common use for speech recognition, e.g. the GMM-HMM and DNN-HMM. While the hidden dynamic models have been evaluated on the large vocabulary tasks involving Switchboard databases, e.g., [12, 66, 68, 86], the RNN has been mainly evaluated on the TIMIT task, e.g., [13, 25, 49, 90].

One particular version of the hidden dynamic model, called the hidden trajectory model, was developed and evaluated after careful design with approximations aimed to overcome the various difficulties associated with localist representations as discussed earlier in this section [38–40]. The main approximation involves using the finite impulse response filter to replace the infinite impulse response one as in the original state equation (6.67) of the state space formulation of the model. This version gives 75.2 % phone recognition accuracy as reported in [38], somewhat higher than 73.9 % obtained by a plain version of the RNN (but with very careful engineering) as reported in Table I of [90, p. 303] and somewhat lower than 76.1 % obtained by an elaborated version of the RNN with LSTM memory units without stacking as reported in Table I of [49, p. 4]. (With less careful engineering, the plain RNN could only achieve 71.8 % accuracy as reported in [25].) This comparison shows that the top-down generative hidden dynamic model based on localist representation of the hidden state performs similarly to the bottom-up discriminative RNN based on distributed representation of the hidden state. This is understandable due to the pros and cons of these different types of models analyzed throughout this section.

6.7 Summary and Discussions on Future Directions

This paper provides an overview on a rather wide range of computational models developed for speech recognition over the past 20 some years. These models are characterized by the use of linear or nonlinear dynamics in the hidden space not directly observed. The temporal unfolding of these dynamic sequence models make the related networks deep, with the depth being the length of the data sequence to be modeled. Among all the models surveyed in this chapter, there are two

fundamentally opposing categories. First, we have the top-down hidden dynamic models of a generative nature. The hidden state adopts the localist representation with explicit physical interpretation and the model parameters are indexed with respect to each of the linguistic/phonetic classes in a parsimonious manner. Second, we have the bottom-up recurrent neural network (RNN) of a discriminative nature. The hidden state adopts the distributed representation with each unit in the hidden state or layer contributing to all linguistic classes.

Sections 6.2 and 6.3 in the early part of this chapter are devoted to the first, generative type of the dynamic models. Section 6.4 describes an interesting class of deep neural network models (DNN) where the network with high depth is constructed independently of the length of the data sequence. In this sense, the DNN technically does not belong to the class of deep dynamic network models discussed above. We include the DNN in this chapter not only due to its prominent role in the current speech recognition practice but also due to the interesting way in which the generative DBN is integrated into the overall DNN learning. In Sect. 6.4, we also discuss how sequence dynamics, an essential part for any sensible speech model, is incorporated into the DNN-based speech model using the HMM as an interface. Section 6.5 then turns to detailed technical reviews on the second type of the (true) dynamic and deep network models for speech, the RNN, which is viewed as a generalization of the DNN where the network's depth is linked to the length of the data sequence.

The most important material of the chapter is Sect. 6.6, which compares the two types of the deep, dynamic models in four incisive aspects. The most critical aspect of the discussion is the localist versus distributed representations for the hidden states, with the respective strengths and weaknesses analyzed in detail. The recognition accuracy achieved by both types of the models is shown to be comparable between the two, implying that the strengths and weaknesses associated with the different model types balance out with each other. (We have analyzed the error patterns and found rather distinct errors produced by the generative hidden dynamic model and by the RNN although the overall error rates are comparable.)

The comprehensive comparisons conducted in Sect. 6.6 shed insights into the question of how to leverage the strengths of both types of models while overcoming their respective weaknesses. Analyzing this future direction is actually the main motivation of this chapter. The integration of the two distinct types of generative and discriminative models may be done blindly as in the case discussed in Sect. 6.4, where the generative DBN is used effectively to initialize or pre-train the discriminative DNN. However, much better strategies can be pursued as present and future directions, given our sufficient understanding by now of the nature of the respective strengths and weaknesses associated with the two model types as elaborated in Sect. 6.6. As an example, one weakness associated with the discriminative RNN, which we briefly mentioned in Sect. 6.6.2, is that distributed representations are not suitable for input to the network. This difficulty has been circumvented in the preliminary work reported in [25] by first using the DNN to extract input features, which gains the advantages of distributed representations embedded in the hidden layers of the DNN. Then the DNN-extracted features equipped with

distributed representations of the data are fed into the subsequent RNN, producing dramatic improvement of phone recognition accuracy from 71.8 % to as high as 81.2 %. Other ways to cleverly get around the problems with localist representations in the generative, deep, and dynamic model and the problems with distributed representations in the discriminative model counterpart are expected to also improve speech recognition performance. As a further example to this end, we also discussed in Sect. 6.6 the strength of the localist representation in easy interpretation of the hidden space of the model. One can take advantage of this strength by using the generative model to create new features that can be effectively combined with other features based on distributed representations. Some advanced approximate inference and learning techniques developed for deep generative models (e.g., [97, 105]) may facilitate successful implementations of this strategy by learning better generative models than the several existing inference and learning methods in the literature (e.g., variational EM and extended Kalman filtering) discussed earlier in this chapter.

References

1. A. Acero, L. Deng, T. Kristjansson, J. Zhang, HMM adaptation using vector taylor series for noisy speech recognition, in *Proceedings of International Conference on Spoken Language Processing* (2000), pp. 869–872
2. J. Baker, Stochastic modeling for automatic speech recognition, in *Speech Recognition*, ed. by D. Reddy (Academic, New York, 1976)
3. J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, D. O'Shgughnessy, Research developments and directions in speech recognition and understanding, part i. IEEE Signal Process. Mag. **26**(3), 75–80 (2009)
4. J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, D. O'Shgughnessy, Updated MINDS report on speech recognition and understanding. IEEE Signal Process. Mag. **26**(4), 78–85 (2009)
5. L. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat. **37**(6), 1554–1563 (1966)
6. Y. Bengio, N. Boulanger, R. Pascanu, Advances in optimizing recurrent networks, in *Proceedings of ICASSP*, Vancouver, 2013
7. Y. Bengio, N. Boulanger-Lewandowski, R. Pascanu, Advances in optimizing recurrent networks, in *Proceedings of ICASSP*, Vancouver, 2013
8. J. Bilmes, Buried markov models: a graphical modeling approach to automatic speech recognition. Comput. Speech Lang. **17**, 213–231 (2003)
9. J. Bilmes, What HMMs can do. IEICE Trans. Inf. Syst. **E89-D**(3), 869–891 (2006)
10. M. Boden, A guide to recurrent neural networks and backpropagation. Tech. rep., T2002:03, SICS (2002)
11. H. Bourlard, N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach. The Kluwer International Series in Engineering and Computer Science*, vol. 247 (Kluwer Academic, Boston, 1994)
12. J. Bridle, L. Deng, J. Picone, H. Richards, J. Ma, T. Kamm, M. Schuster, S. Pike, R. Reagan, An investigation of segmental hidden dynamic models of speech coarticulation for automatic speech recognition. *Final Report for 1998 Workshop on Langauge Engineering, CLSP* (Johns Hopkins, 1998)

13. J. Chen, L. Deng, A primal-dual method for training recurrent neural networks constrained by the echo-state property, in *Proceedings of ICLR* (2014)
14. J.-T. Chien, C.-H. Chueh, Dirichlet class language models for speech recognition. IEEE Trans. Audio Speech Lang. Process. **27**, 43–54 (2011)
15. G. Dahl, D. Yu, L. Deng, A. Acero, Large vocabulary continuous speech recognition with context-dependent DBN-HMMs, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2011)
16. G. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. **20**(1), 30–42 (2012)
17. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum-likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B. **39**, 1–38 (1977)
18. L. Deng, A generalized hidden markov model with state-conditioned trend functions of time for the speech signal. Signal Process. **27**(1), 65–78 (1992)
19. L. Deng, A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition. Speech Commun. **24**(4), 299–323 (1998)
20. L. Deng, Articulatory features and associated production models in statistical speech recognition, in *Computational Models of Speech Pattern Processing* (Springer, New York, 1999), pp. 214–224
21. L. Deng, Computational models for speech production, in *Computational Models of Speech Pattern Processing* (Springer, New York, 1999), pp. 199–213
22. L. Deng, Switching dynamic system models for speech articulation and acoustics, in *Mathematical Foundations of Speech and Language Processing* (Springer, New York, 2003), pp. 115–134
23. L. Deng, *Dynamic Speech Models—Theory, Algorithm, and Applications* (Morgan and Claypool, San Rafael, 2006)
24. L. Deng, M. Aksmanovic, D. Sun, J. Wu, Speech recognition using hidden Markov models with polynomial regression functions as non-stationary states. IEEE Trans. Acoust. Speech Signal Process. **2**(4), 101–119 (1994)
25. L. Deng, J. Chen, Sequence classification using high-level features extracted from deep neural networks, in *Proceedings of ICASSP* (2014)
26. L. Deng, J. Droppo, A. Acero, A Bayesian approach to speech feature enhancement using the dynamic cepstral prior, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1 (2002), pp. I-829–I-832
27. L. Deng, K. Hassanein, M. Elmasry, Analysis of the correlation structure for a neural predictive model with application to speech recognition. Neural Netw. **7**(2), 331–339 (1994)
28. L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related applications: an overview, in *Proceedings of IEEE ICASSP*, Vancouver, 2013
29. L. Deng, G. Hinton, D. Yu, Deep learning for speech recognition and related applications, in *NIPS Workshop*, Whistler, 2009
30. L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, P. Mermelsten, Phonemic hidden markov models with continuous mixture output densities for large vocabulary word recognition. IEEE Trans. Acoust. Speech Signal Process. **39**(7), 1677–1681 (1991)
31. L. Deng, L. Lee, H. Attias, A. Acero, Adaptive kalman filtering and smoothing for tracking vocal tract resonances using a continuous-valued hidden dynamic model. IEEE Trans. Audio Speech Lang. Process. **15**(1), 13–23 (2007)
32. L. Deng, M. Lennig, F. Seitz, P. Mermelstein, Large vocabulary word recognition using context-dependent allophonic hidden markov models. Comput. Speech Lang. **4**, 345–357 (1991)
33. L. Deng, X. Li, Machine learning paradigms in speech recognition: an overview. IEEE Trans. Audio Speech Lang. Process. **21**(5), 1060–1089 (2013)
34. L. Deng, J. Ma, A statistical coarticulatory model for the hidden vocal-tract-resonance dynamics, in *EUROSPEECH* (1999), pp. 1499–1502

35. L. Deng, J. Ma, Spontaneous speech recognition using a statistical coarticulatory model for the hidden vocal-tract-resonance dynamics. J. Acoust. Soc. Am. **108**, 3036–3048 (2000)
36. L. Deng, D. O'Shaughnessy, *Speech Processing—A Dynamic and Optimization-Oriented Approach* (Marcel Dekker, New York, 2003)
37. L. Deng, G. Ramsay, D. Sun, Production models as a structural basis for automatic speech recognition. Speech Commun. **33**(2–3), 93–111 (1997)
38. L. Deng, D. Yu, Use of differential cepstra as acoustic features in hidden trajectory modelling for phonetic recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2007), pp. 445–448
39. L. Deng, D. Yu, A. Acero, A bidirectional target filtering model of speech coarticulation: two-stage implementation for phonetic recognition. IEEE Trans. Speech Audio Process. **14**, 256–265 (2006)
40. L. Deng, D. Yu, A. Acero, Structured speech modeling. IEEE Trans. Speech Audio Process. **14**, 1492–1504 (2006)
41. P. Divenyi, S. Greenberg, G. Meyer, *Dynamics of Speech Production and Perception* (IOS Press, Amsterdam, 2006)
42. J. Droppo, A. Acero, Noise robust speech recognition with a switching linear dynamic model, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1 (2004), pp. I-953–I-956
43. E. Fox, E. Sudderth, M. Jordan, A. Willsky, Bayesian nonparametric methods for learning markov switching processes. IEEE Signal Process. Mag. **27**(6), 43–54 (2010)
44. B. Frey, L. Deng, A. Acero, T. Kristjansson, Algonquin: iterating laplaces method to remove multiple types of acoustic distortion for robust speech recognition, in *Proceedings of Eurospeech* (2000)
45. M. Gales, S. Young, Robust continuous speech recognition using parallel model combination. IEEE Trans. Speech Audio Process. **4**(5), 352–359 (1996)
46. Z. Ghahramani, G.E. Hinton, Variational learning for switching state-space models. Neural Comput. **12**, 831–864 (2000)
47. Y. Gong, I. Illina, J.-P. Haton, Modeling long term variability information in mixture stochastic trajectory framework, in *Proceedings of International Conference on Spoken Language Processing* (1996)
48. A. Graves, Sequence transduction with recurrent neural networks, in *Representation Learning Workshop, ICML* (2012)
49. A. Graves, A. Mahamed, G. Hinton, Speech recognition with deep recurrent neural networks, in *Proceedings of ICASSP*, Vancouver, 2013
50. G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in Technical report 2010-003, Machine Learning Group, University of Toronto, 2010.
51. G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
52. G. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
53. G. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
54. W. Holmes, M. Russell, Probabilistic-trajectory segmental HMMs. Comput. Speech Lang. **13**, 3–37 (1999)
55. X. Huang, A. Acero, H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development* (Upper Saddle River, New Jersey 07458)
56. H. Jaeger, Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. GMD Report 159, GMD - German National Research Institute for Computer Science (2002)
57. F. Jelinek, Continuous speech recognition by statistical methods. Proc. IEEE **64**(4), 532–557 (1976)

58. B.-H. Juang, S.E. Levinson, M.M. Sondhi, Maximum likelihood estimation for mixture multivariate stochastic observations of markov chains. IEEE Trans. Inf. Theory **32**(2), 307–309 (1986)
59. B. Kingsbury, T. Sainath, H. Soltau, Scalable minimum Bayes risk training of deep neural network acoustic models using distributed hessian-free optimization, in *Proceedings of Interspeech* (2012)
60. H. Larochelle, Y. Bengio, Classification using discriminative restricted Boltzmann machines, in *Proceedings of the 25th International Conference on Machine learning* (ACM, New York, 2008), pp. 536–543
61. L. Lee, H. Attias, L. Deng, Variational inference and learning for segmental switching state space models of hidden speech dynamics, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1 (2003), pp. I-872–I-875
62. L.J. Lee, P. Fieguth, L. Deng, A functional articulatory dynamic model for speech production, in *Proceedings of ICASSP*, Salt Lake City, vol. 2, 2001, pp. 797–800
63. S. Liu, K. Sim, Temporally varying weight regression: a semi-parametric trajectory model for automatic speech recognition. IEEE Trans. Audio Speech Lang. Process. **22**(1) 151–160 (2014)
64. S.M. Siniscalchia, D. Yu, L. Deng, C.-H. Lee, Exploiting deep neural networks for detection-based speech recognition. Neurocomputing **106**, 148–157 (2013)
65. J. Ma, L. Deng, A path-stack algorithm for optimizing dynamic regimes in a statistical hidden dynamic model of speech. Comput. Speech Lang. **14**, 101–104 (2000)
66. J. Ma, L. Deng, Efficient decoding strategies for conversational speech recognition using a constrained nonlinear state-space model. IEEE Trans. Audio Speech Process. **11**(6), 590–602 (2003)
67. J. Ma, L. Deng, Efficient decoding strategies for conversational speech recognition using a constrained nonlinear state-space model. IEEE Trans. Audio Speech Lang. Process **11**(6), 590–602 (2004)
68. J. Ma, L. Deng, Target-directed mixture dynamic models for spontaneous speech recognition. IEEE Trans. Audio Speech Process. 12(1), 47–58 (2004)
69. A.L. Maas, Q. Le, T.M. O'Neil, O. Vinyals, P. Nguyen, A.Y. Ng, Recurrent neural networks for noise reduction in robust asr, in *Proceedings of INTERSPEECH*, Portland, 2012
70. J. Martens, I. Sutskever, Learning recurrent neural networks with hessian-free optimization, in *Proceedings of ICML*, Bellevue, 2011, pp. 1033–1040
71. G. Mesnil, X. He, L. Deng, Y. Bengio, Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding, in *Proceedings of INTERSPEECH*, Lyon, 2013
72. B. Mesot, D. Barber, Switching linear dynamical systems for noise robust speech recognition. IEEE Trans. Audio Speech Lang. Process. **15**(6), 1850–1858 (2007)
73. T. Mikolov, Statistical language models based on neural networks, Ph.D. thesis, Brno University of Technology, 2012
74. T. Mikolov, A. Deoras, D. Povey, L. Burget, J. Cernocky, Strategies for training large scale neural network language models, in *Proceedings of IEEE ASRU* (IEEE, Honolulu, 2011), pp. 196–201
75. T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, S. Khudanpur, Recurrent neural network based language model, in *Proceedings of INTERSPEECH*, Makuhari, 2010, pp. 1045–1048
76. T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, S. Khudanpur, Extensions of recurrent neural network language model, in *Proceedings of IEEE ICASSP*, Prague, 2011, pp. 5528–5531
77. A. Mohamed, G. Dahl, G. Hinton, Acoustic modeling using deep belief networks. IEEE Trans. Audio Speech Lang. Process. **20**(1), 14–22 (2012)
78. A. Mohamed, G.E. Dahl, G.E. Hinton, Deep belief networks for phone recognition, in *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications* (2009)
79. A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, M. Picheny, Deep belief networks using discriminative features for phone recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2011), pp. 5060–5063

80. N. Morgan, Deep and wide: multiple layers in automatic speech recognition. IEEE Trans. Audio Speech Lang. Process. **20**(1), 7–13 (2012)
81. M. Ostendorf, V. Digalakis, O. Kimball, From HMM's to segment models: a unified view of stochastic modeling for speech recognition. IEEE Trans. Speech Audio Process. **4**(5), 360–378 (1996)
82. M. Ostendorf, A. Kannan, O. Kimball, J. Rohlicek, Continuous word recognition based on the stochastic segment model, in *Proceedings of DARPA Workshop CSR* (1992)
83. E. Ozkan, I. Ozbek, M. Demirekler, Dynamic speech spectrum representation and tracking variable number of vocal tract resonance frequencies with time-varying dirichlet process mixture models. IEEE Trans. Audio Speech Lang. Process. **17**(8), 1518–1532 (2009)
84. R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in *Proceedings of ICML*, Atlanta, 2013
85. V. Pavlovic, B. Frey, T. Huang, Variational learning in mixed-state dynamic graphical models, in *Proceedings of UAI*, Stockholm, 1999, pp. 522–530
86. J. Picone, S. Pike, R. Regan, T. Kamm, J. Bridle, L. Deng, Z. Ma, H. Richards, M. Schuster, Initial evaluation of hidden dynamic models on conversational speech, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (1999)
87. G. Puskorius, L. Feldkamp, Neurocontrol of nonlinear dynamical systems with kalman filter trained recurrent networks. IEEE Trans. Neural Netw. **5**(2), 279–297 (1998)
88. L. Rabiner, B.-H. Juang, *Fundamentals of Speech Recognition* (Prentice-Hall, Upper Saddle River, 1993)
89. S. Rennie, J. Hershey, P. Olsen, Single-channel multitalker speech recognition—graphical modeling approaches. IEEE Signal Process.Mag. **33**, 66–80 (2010)
90. A.J. Robinson, An application of recurrent nets to phone probability estimation. IEEE Trans. Neural Netw. **5**(2), 298–305 (1994)
91. A. Rosti, M. Gales, Rao-blackwellised gibbs sampling for switching linear dynamical systems, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1 (2004), pp. I-809–I-812
92. M. Russell, P. Jackson, A multiple-level linear/linear segmental HMM with a formant-based intermediate layer. Comput. Speech Lang. **19**, 205–225 (2005)
93. T. Sainath, B. Kingsbury, H. Soltau, B. Ramabhadran, Optimization techniques to improve training speed of deep neural networks for large speech tasks. IEEE Trans. Audio Speech Lang. Process. **21**(11), 2267–2276 (2013)
94. F. Seide, G. Li, X. Chen, D. Yu, Feature engineering in context-dependent deep neural networks for conversational speech transcription, in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2011* (Waikoloa, HI, USA), pp. 24–29
95. X. Shen, L. Deng, Maximum likelihood in statistical estimation of dynamical systems: decomposition algorithm and simulation results. Signal Process. **57**, 65–79 (1997)
96. K. N. Stevens, Acoustic phonetics, Vol. 30, MIT Press, 2000.
97. V. Stoyanov, A. Ropson, J. Eisner, Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure, in *Proceedings of AISTAT* (2011)
98. I. Suskever, J. Martens, G.E. Hinton, Generating text with recurrent neural networks, in *Proceedings of 28th International Conference on Machine Learning* (2011)
99. I. Sutskever, Training recurrent neural networks, Ph.D. thesis, University of Toronto, 2013
100. I. Sutskever, J. Martens, G.E. Hinton, Generating text with recurrent neural networks, in *Proceedings of ICML*, Bellevue, 2011, pp. 1017–1024
101. R. Togneri, L. Deng, Joint state and parameter estimation for a target-directed nonlinear dynamic system model. IEEE Trans. Signal Process. **51**(12), 3061–3070 (2003)
102. R. Togneri, L. Deng, A state-space model with neural-network prediction for recovering vocal tract resonances in fluent speech from mel-cepstral coefficients. Speech Commun. **48**(8), 971–988 (2006)
103. F. Triefenbach, A. Jalalvand, K. Demuynck, J.-P. Martens, Acoustic modeling with hierarchical reservoirs. EEE Trans. Audio Speech Lang. Process. **21**(11), 2439–2450 (2013)

104. S. Wright, D. Kanevsky, L. Deng, X. He, G. Heigold, H. Li, Optimization algorithms and applications for speech and language processing. IEEE Trans. Audio Speech Lang. Process. **21**(11), 2231–2243 (2013)
105. X. Xing, M. Jordan, S. Russell, A generalized mean field algorithm for variational inference in exponential families, in *Proceedings of UAI* (2003)
106. D. Yu, L. Deng, Speaker-adaptive learning of resonance targets in a hidden trajectory model of speech coarticulation. Comput. Speech Lang. **27**, 72–87 (2007)
107. D. Yu, L. Deng, Discriminative pretraining of deep neural networks, US Patent 20130138436 A1, 2013
108. D. Yu, L. Deng, G. Dahl, Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition, in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning* (2010)
109. D. Yu, F. Seide, G. Li, L. Deng, Exploiting sparseness in deep neural networks for large vocabulary speech recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2012), pp. 4409–4412
110. D. Yu, S. Siniscalchi, L. Deng, C. Lee, Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2012)
111. H. Zen, K. Tokuda, T. Kitamura, An introduction of trajectory model into HMM-based speech synthesis, in *Proceedings of ISCA SSW5* (2004), pp. 191–196
112. L. Zhang, S. Renals, Acoustic-articulatory modelling with the trajectory HMM. IEEE Signal Process. Lett. **15**, 245–248 (2008)

Chapter 7
Speech Based Emotion Recognition

Vidhyasaharan Sethu, Julien Epps, and Eliathamby Ambikairajah

Abstract This chapter will examine current approaches to speech based emotion recognition. Following a brief introduction that describes the current widely utilised approaches to building such systems, it will attempt to broadly segregate components commonly involved in emotion recognition systems based on their function (i.e., feature extraction, normalisation, classifier, etc.) to give a broad view of the landscape. The next section of the chapter will then attempt to explain in more detail those components that are part of the most current systems. The chapter will also present a broad overview of how phonetic and speaker variability are dealt with in emotion recognition systems. Finally, the chapter presents the authors' views on what are the current and future research challenges in the field.

7.1 Introduction

Speech has played a significant role in the evolution of humans and is probably the most natural and widely used form of interpersonal communication. While in general the primary objective of speech is to convey information encoded as linguistic content (via one of numerous languages), speech is not completely characterised by its linguistic content. Factors such as the speaker's anatomical and physiological traits, behavioural traits, emotional state, mental state, and cognitive state also influence speech characteristics and are collectively referred to as paralinguistic content. Humans are able to both convey and interpret paralinguistic information in speech with very little effort during the course of any normal conversation (Fig. 7.1).

V. Sethu (✉)
School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, NSW, Australia
e-mail: v.sethu@unsw.edu.au

J. Epps • E. Ambikairajah
School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, NSW, Australia

ATP Research Laboratory, National ICT Australia (NICTA), Eveleigh, NSW, Australia
e-mail: ambi@ee.unsw.edu.au

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_7

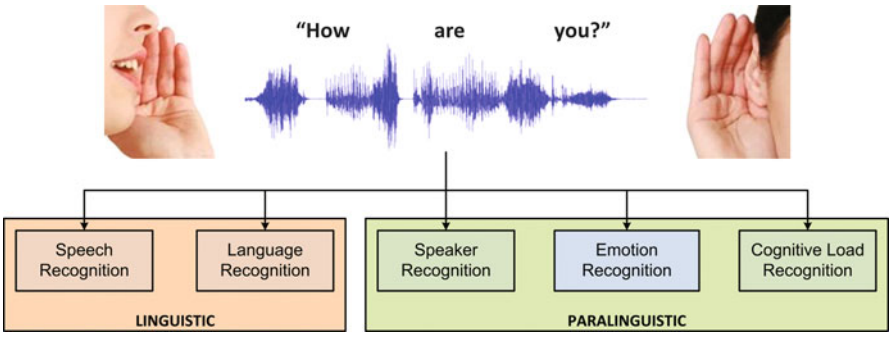

Fig. 7.1 Speech based information recognition problems

Research into recognition of paralinguistic information from speech, specifically emotion recognition systems, has been extremely active for more than a decade now. However, emotion recognition systems are still not 'mature' in the way modern speech and speaker recognition systems are. One of several possible reasons for this state of affairs is probably that even the definition of emotions is not a scientifically settled issue. Given this state of affairs, it is quite possible that emotion recognition systems of the future could be very different from the various approaches that are state-of-the-art today. However, state-of-the-art systems today do share a number of common trends and the aim of this chapter is to elaborate on the common approaches taken by current emotion recognition systems.

7.1.1 What Are Emotions?

Emotions are specific and consistent collections of physiological responses triggered by internal or external representations of certain objects or situations, such as a change in the person's body that produces pain, or an external stimulus such as the sight of another person; or the representation, from memory, of a person, or object, or situation in the thought process. There is some evidence to suggest that the basics of most if not all emotional responses are preset in the genome [1]. In a broad sense, emotions are a part of the bio-regulatory mechanism that humans have evolved to maintain life and survive. Emotions form an intermediary layer between stimulus and behavioural reaction, replacing rigid reflex-like response patterns [2] allowing for greater flexibility in behaviour [3]. Emotional reactions also serve as a signalling system between organisms and are essential in acquiring new behaviour patterns. It has been pointed out that they are a pre-requisite for learning [4, 5]. It should also be noted that while emotions are often referred to as 'states', they are in fact not static concepts but constantly changing processes. Further, since emotions may result in significant utilisation of biological resources, emotional states may be expected to be of relatively short duration [6]. While emotion durations can vary

greatly (for e.g., a person may experience a few seconds of anger or a day of anger), survey based studies have found that in general emotion durations are typically in the order of a few minutes and less than an hour [7, 8].

One of the most widely accepted frameworks for defining emotions is the component process model [2, 6], which frames emotions as psychological constructs consisting of synchronised changes in the states of five subsystems in response to external or internal stimuli. The term 'components' refers to the states of the subsystems, each serving a distinct function (listed below), and the term 'process' to the coordinated changes to the states over time that constitute an emotion.

Component	Function
Cognitive stimuli appraisal	Evaluation of an environment
Neurophysiological processes	System regulation
Motivational and behavioural tendencies	Preparation of action
Motor expression	Communication of intention
Subjective feeling	Reflection and monitoring

Based on internal and external stimuli, the state of each of the components is continuously changing (e.g., the sight of a desirable object will change state of the cognitive stimuli appraisal component from 'seeing an object' to 'evaluating it as desirable'; and the state of the motivational component from 'curious' to 'wanting the object' and so on). An emotion is then conceptualised as a pattern of state changes in these components where each component is influenced by the others [2].

A systemic approach to develop a theory of emotions leads to 'appraisal theories of emotion', all of which fit a component-process framework. An overview of appraisal theories of emotion can be found in [9]. In light of these theories it has been suggested that automatic emotion recognition should be carried out as appraisal classification, followed by mapping appraisals to emotions. However this approach is yet to be realised.

Based on the definition of emotions as including a physiological component, both voluntary and involuntary effects on the human speech production apparatus can be expected and the characteristics of vocal expression are the net result of these effects. It has been noted that characteristics affecting bodily movement also affect the voice production mechanism and consequently the voice. This is supported by the observation that the vocal expressions of basic emotions is similar in many languages [10]. This work also notes considerable parallels between vocal and physical gestures—for example, an increased tension in the throat causing an increased loudness of speech paralleling an increased tension of the whole body in preparation for an imminent fight. An even more innate 'frequency code' with high frequency vocalisation suggesting a submissive attitude and lower frequency vocalisation suggesting greater size and a more dominant attitude was proposed in [11]. Demonstrations suggesting that various aspects of a speaker's physical and emotional state, including age, sex and personality can be identified by voice alone are reviewed in [12]. This low-level information is present in even short utterances and could influence the interpretation of the words being uttered, typically identified

by "it's not what he said but the way he said it". An analogy from communication interprets the paralinguistic information as an "emotion carrier wave" for the words [13]. Consequently, emotion can still be recognised even if the linguistic information is not interpreted, this is further supported by the work reported in [14] noting that emotion can be recognised from segments of speech as short as 60 ms. Scherer et al. [15] report an emotion recognition accuracy of 66 % on meaningless multilingual sentences by listeners from different cultural backgrounds, and interpret this as evidence for the existence of vocal characteristics specific to emotions.

Various other authors have also hinted at systematic correlations between emotions and acoustic parameters [16–19]. It should be noted that the relationships that have been reported in literature are not always consistent across all studies and may contradict each other. However, most relationships are consistent and point towards correlations between emotions and acoustic parameters that can be exploited by an automatic emotion recognition system. To illustrate some emotion-specific effects, spectrograms corresponding to the word 'Thousand' spoken by the same person without expressing any emotion (Fig. 7.2a); and while expressing anger (Fig. 7.2b) are shown below. Reviews of research investigating the effect of emotions on vocal expression can be found in [13, 20–23].

7.1.2 Emotion Labels

One consequence of not having a fully established theory of emotions is that the question of how to label emotions does not have a straightforward answer. Human languages contain a large number of 'emotion denoting' adjectives. According to Cowie and Cornelius [24], the Semantic Atlas of Emotion Concepts [25] lists 558 words with 'emotional connotations'. However, it may be that not all of these terms are equally important and given the specific research aims it could be possible to select a subset of these terms fulfilling certain requirements. A number of such approaches have been proposed including: basic emotions from a Darwinian point of view, which are shaped by evolution to serve functions that benefit survival [26]; emotion categories chosen on the grounds that they are more fundamental than others because they encompass the other emotion categories; and asking people what emotion terms play an important role in everyday life [27].

While the aim of the above mentioned approaches is to reduce the number of emotion related terms, it has also been argued that emotions are a continuum and these terms, even a very large number of them, do not capture every shade of emotion a person can distinguish (even though people would naturally describe emotional experiences in these categorical terms). The dimensional approach to emotion categorisation is also related to this line of argument in that it describes shades of emotions as points in a continuous two- or three- dimensional space. Some of the earliest (modern) studies involving the dimensional description of emotions are reported in [28, 29], and in [23], emotional states are described in terms of a two-dimensional circular space, with axes labelled 'activation' (going

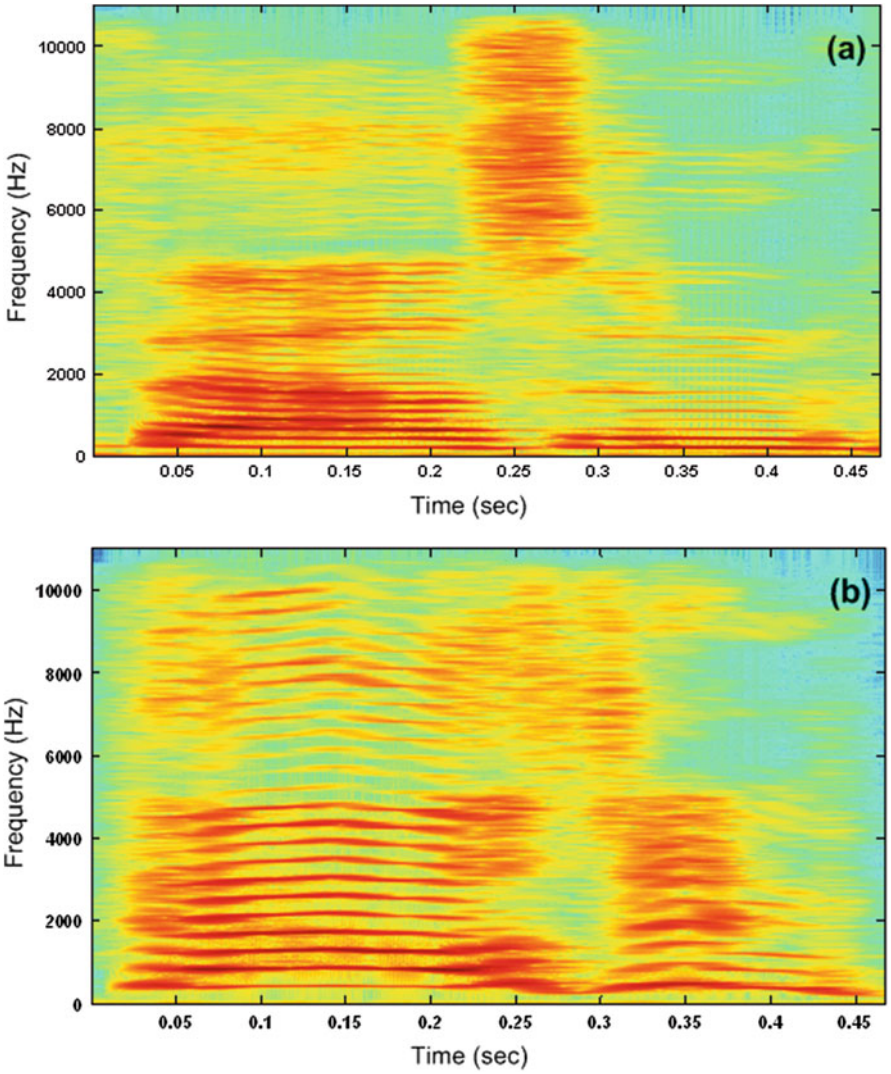

Fig. 7.2 Spectrograms corresponding to the word 'Thousand' (Emotional Prosody Speech and Transcripts database) spoken while expressing (**a**) no emotion and (**b**) anger

from passive to active) and 'evaluation' or 'valence' (going from negative to positive). An important question with the dimensional approach is then whether these emotion dimensions capture all relevant properties of the emotion concepts or if they are simplified and reduced descriptions. Opinion is once again divided, with Russel et al. [30] claiming that three dimensions emerging from their factor analysis is 'sufficient to define all the various emotional states', while the opposite view is expressed in [31]. More comprehensive overviews of various descriptive frameworks can be found in [24, 32].

Among the different labelling paradigms, the use of categorical labels would be the one most familiar from everyday life. It is also the most common approach taken by automatic emotion recognition systems. The subset of categorical emotion labels that are most frequently used are the following, commonly referred to as the 'Big six' [33, 34].

1. Happiness
2. Sadness
3. Fear
4. Disgust
5. Anger
6. Surprise

An observation that can be made at this point is that, regardless of whether emotions are labelled with discrete categorical terms or along continuous axes, the problem of labelling emotions can be concretely formulated. This in turn brings automatic emotion recognition systems into the realm of possibility.

7.1.3 The Emotion Recognition Task

Given the state of affairs with no settled theory of emotions or even universal agreement on how emotions should be labelled, it should come as no surprise that there is no single emotion recognition task. Broadly speaking, the term 'emotion recognition' can refer to either a regression problem or to a classification problem. Regression systems take a dimensional approach to labelling emotions and aim to estimate scores along each dimension [35–38]. Systems that take a classification approach may use either dimensional labelling or discrete labelling for emotions, and can be further categorised into multi-class and two-class systems. Two class systems may be either those that determine presence vs. absence of a particular emotion [39], or those distinguishing between two extremes of a dimension used to quantify emotion, such as high vs. low arousal or positive vs. negative valence [40, 41] (Fig. 7.3).

Over the years, a number of different approaches have been taken in the development of the different kinds of emotion recognition systems listed above. The good news is that at a high level, the different approaches to the classification and regression problems have converged to a similar overall scheme, in terms of feature extraction and classification/regression, for all systems which is the focus of this chapter.

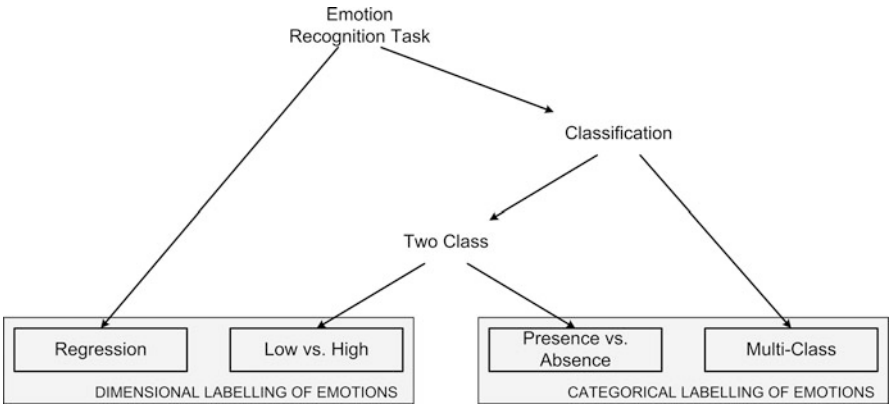

Fig. 7.3 Overview of automatic emotion recognition problems

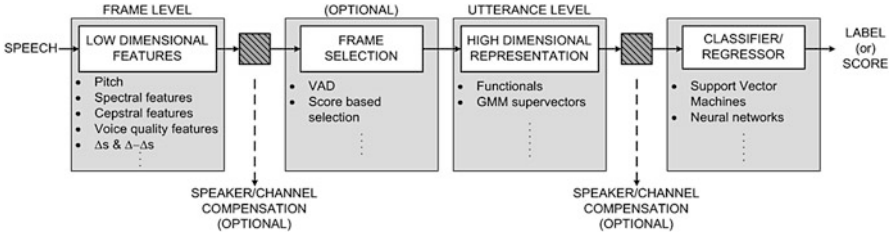

Fig. 7.4 General high-level structure of state-of-the-art emotion recognition system

7.2 Emotion Classification Systems

Most state-of-the-art emotion classification systems are functionally similar at a high level but can differ quite significantly at a lower level. The first stage generally involves extraction of low-level features such as MFCCs, pitch, etc. from short frames (in the order of a few tens of milliseconds) of speech. Following this, a high dimensional representation of the set of all short term frame based feature vectors from an utterance is estimated. In most approaches, a VAD, voicing detector or a similar frame selection process is used and the high dimensional representation is estimated from some subset of all the short term frame based feature vectors. Finally, an appropriately trained classifier or regressor in the back-end operates on this high dimensional representation of a speech utterance in order to determine emotional class, or estimates of dimensional scores. In addition to this basic setup, most systems also have additional components for speaker normalisation and/or channel compensation (Fig. 7.4).

7.2.1 Short-Term Features

Information about emotional state is expressed via speech through numerous cues, ranging from low-level acoustic ones to high-level linguistic content. Several approaches to speech based automatic emotion recognition, each taking advantage of a few of these cues, have been explored [42–50]. It would be impossible to list all of them, however, approaches that use linguistic cues [51–53] are not as common as those that make use of low-level acoustic and prosodic cues due to the complexity inherent in the accurate estimation of linguistic cues and their language-dependence.

Features representing the common low-level acoustic and prosodic cues are generally extracted from short-term frames (approximately 20–40 ms). In many cases, the deltas (Δs), and delta–deltas ($\Delta\Delta$s) of these features are appended to them to form the final feature vector. Deltas (Δs) are element-wise first order temporal differences and delta–deltas ($\Delta\Delta s$) element-wise deltas of deltas, i.e., second order temporal differences. In addition, other temporal derivatives such as shifted delta coefficients (SDCs) and deltas based on regression (i.e., slope estimated from a number of frames on either side of the current frame, instead of just one on either side) are also utilised in some systems. Short term features commonly used in speech based emotion recognition systems can be grouped in the following broad categories:

7.2.1.1 Pitch

Technically the term pitch refers to the fundamental frequency as perceived by a receiver but in the context of feature extraction it is almost universally used to refer to the actual fundamental frequency (F_0). Typically a single pitch value is determined from each frame and this may be followed up with some form of post-processing. Pitch has been shown to play an important role in emotion recognition [54, 55] and a range of prosodic parameters related to it such as pitch level, pitch range and jitter have been widely used in emotion recognition systems.

7.2.1.2 Loudness/Energy

The intensity of speech is a measure of the energy contained in speech as it is produced, which in turn is based on the energy of the vocal excitation. Along with pitch, loudness based features are the most common prosodic features used in emotion recognition systems. Loudness of speech as perceived by a listener on the other hand depends on the sound pressure level (SPL) of the sound waves at the eardrum (or microphone), which is dependent on both the intensity of the speech and the distance of between the speaker and the listener (microphone). Consequently, when using loudness as a measure of vocal excitation intensity, a possibly unrealistic assumption that the speakers are always at the same distance from the microphone is being made.

7.2.1.3 Spectral Features

A number of features may be employed to capture short term spectral characteristics of speech. These range from single dimensional representations of spectra such as zero crossing rate, spectral centroid, spectral slope, etc. to multi-dimensional representations such as a vector of formant frequencies and formant magnitudes, modulation features, etc.

7.2.1.4 Cepstral Features

Cepstral features are similar to spectral features in that they capture short term spectral characteristics of speech as well. Cepstral features such as MFCC features are the most widely used short term features in speech based emotion recognition systems. Cepstral features tend to differ from spectral features in a couple of ways. Firstly, cepstral features are generally reasonably detailed (multidimensional) descriptions of the short term spectrum, while spectral features may be broad or detailed. Secondly, cepstral feature dimensions are far less correlated with each other compared to detailed spectral feature dimensions. This reduced correlation makes them particularly suited for building Gaussian mixture models (GMMs) and hidden Markov models (HMMs) where each state is modelled by a GMM.

7.2.2 High Dimensional Representation

The short term features by definition are incapable of capturing information about long-term temporal evolution of speech parameters. Even with the inclusion of delta features (and its variants), each short-term feature vector represents less than a tenth of a second of speech. This shortcoming is widely recognised and automatic emotion recognition systems generally deal with it in one of two ways. One approach is to use appropriate back-ends, which are capable of modelling patterns of sequences of features vectors, such as hidden Markov models, or neural networks with memory to model temporal patterns of the short-term features. This approach is outlined in Sect. 7.2.4. The more widely used approach however is to estimate a sequence of short-term feature vectors from each utterance and then compute a high dimensional representation of the utterance from the sequence of short-term feature vectors. The high dimensional representation is generally designed to be representative of the statistical properties of short-term features across the entire utterance and take into account long-term temporal patterns. It should be noted that in this approach information about the long-term temporal patterns may not be captured if the high-dimensional representation is not appropriately designed.

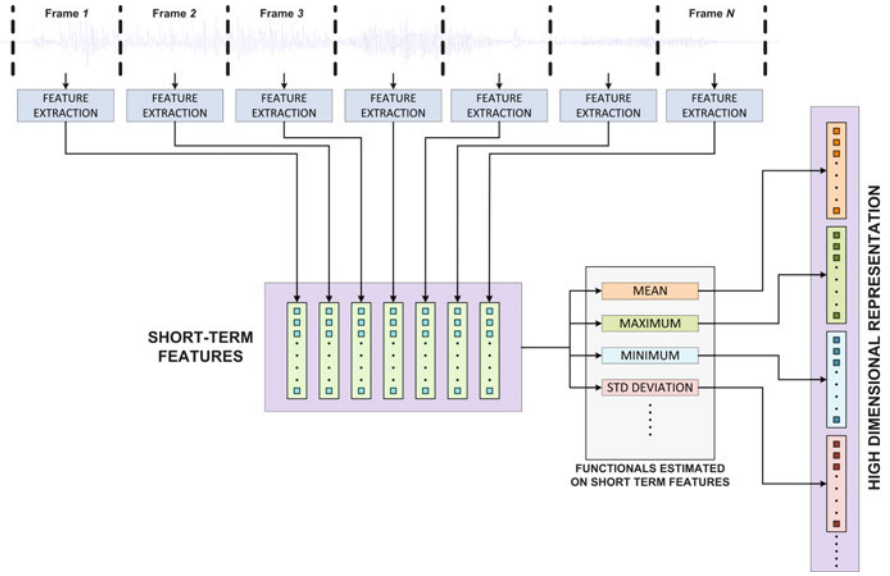

Fig. 7.5 Estimating high dimensional representation of an utterance using functionals (typically statistical descriptors)

7.2.2.1 Functional Approach to a High-Dimensional Representation

The most widely used method for estimating a high dimensional representation of an utterance involves evaluating a range of appropriate functionals (typically statistical descriptors) of the different dimensions of the short-term features (see Fig. 7.5) [56]. As an example, extracting 13-dimensional MFCC feature vectors using 20 ms frames with 50 % overlap from an utterance of duration 2 s results in 199 vectors of 13-dimensions. Functionals may then be evaluated on each of the 13 dimensions across the 199 values (such as the maximum value of the first MFCC element across all 199 feature vectors) to form each element of a high-dimensional representation. Hence, if N functionals are evaluated on a set of T feature vectors of D dimensions, the resultant high-dimensional representation is an ND dimensional vector. It is important to note that the dimensionality of this vector is independent of utterance length (and consequently the number of frames, T). This essential property simplifies classifier/regressor design since all decisions are made based on a single fixed dimensional vector.

Given a sufficiently large number of functionals, the high-dimensional representation can easily span tens or hundreds of thousands of dimensions. This is generally followed by a dimensionality reduction stage where typically feature selection algorithms are employed to identify and extract the most important dimensions. An overview comparing a number of short-term features and functionals utilised in current state-of-the-art systems can be found in [57]. The 'traditional' approach

to building speech based classification systems have typically involved handpicking/designing low-dimensional features based on prior knowledge of the problem domain. The approach outlined in this section automates this process of manually selecting and evaluating features from a large pool of possible descriptions.

7.2.2.2 GMM Supervector Approach to High-Dimensional Representation

An alternative approach to obtaining statistical descriptions of short-term features is to estimate the underlying probability distribution of the short-term features and use a parametric representation of this distribution. Most state-of-the-art speaker identification and language identification systems take this approach and utilise Gaussian mixture models (GMMs) as parametric models of feature distributions [58, 59]. Generally MFCC $+ \Delta + \Delta\Delta$ (or similar features) which provide detailed descriptions of short-term spectra while containing relatively decorrelated feature dimension are the most commonly modelled short-term features. Given that Gaussian mixture models are powerful representations of feature distributions and given the wealth of variability compensation and discriminative training techniques available for use with GMM systems, the use of GMM supervectors to represent utterances in emotion recognition systems is a promising avenue of investigation. It should also be noted that this approach promises competitive performance despite predominantly using only cepstral features [60].

In general, GMM supervectors are estimated by first training a Gaussian mixture model, referred to as a universal background model (UBM), on generic speech representative of the style that is expected to be confronted by the operating emotion recognition system and preferably from many different speakers if the emotion recognition system is expected to work with multiple speakers. This UBM is then adapted (via MAP adaptation) towards each utterance to obtain an estimate of a model of the feature distributions corresponding to them. A GMM supervector is a vectorial representation of the parameters of these models. In the common case where only the UBM means are adapted via MAP adaptation, the GMM supervector is composed by concatenating the means of each mixture component of the adapted GMM (see Fig. 7.6).

A number of variability compensation techniques that have been developed for GMM based systems in the context of speaker and language identification tasks may be utilised to compensate for undesired feature variability due to non-emotional factors such as the linguistic content (what is being said) and speaker differences (who is saying it). Some of these variability compensation methods are outlined in Sect. 7.3. It should be noted that while GMM supervector based system constitute the state-of-the-art in speaker identification and language identification, most current emotion recognition systems utilise classification systems based on high-dimensional representation composed of functionals evaluated on short-term features as outlined in Sect. 7.2.2.1.

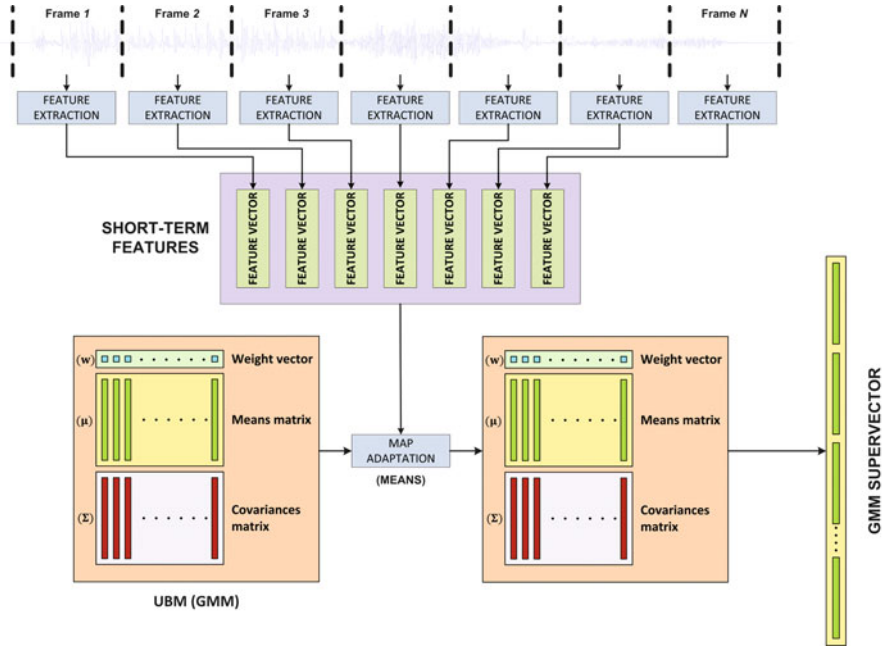

Fig. 7.6 Estimating GMM supervectors as a high dimensional representation

7.2.3 Modelling Emotions

Current literature contains examples of a variety of classifiers that have been employed by emotion recognition systems. Based on their approach, these classifiers may be either generative models or discriminative functions. Generative classifiers try to model the distribution of the training data (features) from each class (emotion) individually (i.e., the models of each class are based only on data from that class and not from any other class). Pattern matching involves estimating some measure of closeness of the unknown data to each of the models and then selecting the class whose model is closest to the data. Some generative classifiers used in emotion recognition are:

- Gaussian mixture models (GMM) [61–63]
- Hidden Markov models (HMM) [63–66]
- Probabilistic neural networks [63, 67]

Unlike generative classifiers, which attempt to model the entire feature space for each class, discriminative classifiers attempt to maximise a discriminative function between the different classes without modelling the distribution of the entire feature space. Examples of used discriminative classifiers commonly utilised in emotion recognition systems are:

- Support vector machines (SVM) [63, 68–71]
- Decision trees and Random forests (ensemble of tress) [72–74]
- Neural networks (NN) [56, 75, 76]

In general, since generative classifiers attempt to model the probability distributions of the feature space, they all tend to suffer from the curse of dimensionality and are not suitable for modelling high dimensional feature spaces. Among the discriminative classifiers, by far the most commonly utilised choice for emotion classification based on high dimensional representations is support vector machines, owing to the nature of their operation, the existence of efficient training methodologies and the large number of kernels that they may utilise to match the properties of the feature space. They have been shown to outperform other common classifiers in some emotion recognition problems [77].

7.2.3.1 Emotion Models: Linear Support Vector Machines

Fundamentally, support vector machines are binary (two-class) classifiers. Multi-class classification problems are tackled by decomposing the problem into a number of two-class problems, each of which is then addressed by a support vector machine. Training a linear SVM (conceptually non-linear SVMs are an extension to linear SVMs and will be discussed later) involves estimating the hyperplane in the feature space that best separates training feature vectors of the two classes in terms of 'support vectors' (shown in two feature dimensions in Fig. 7.7). Support vectors are feature vectors from the training set (of either class) that are close to the best linear

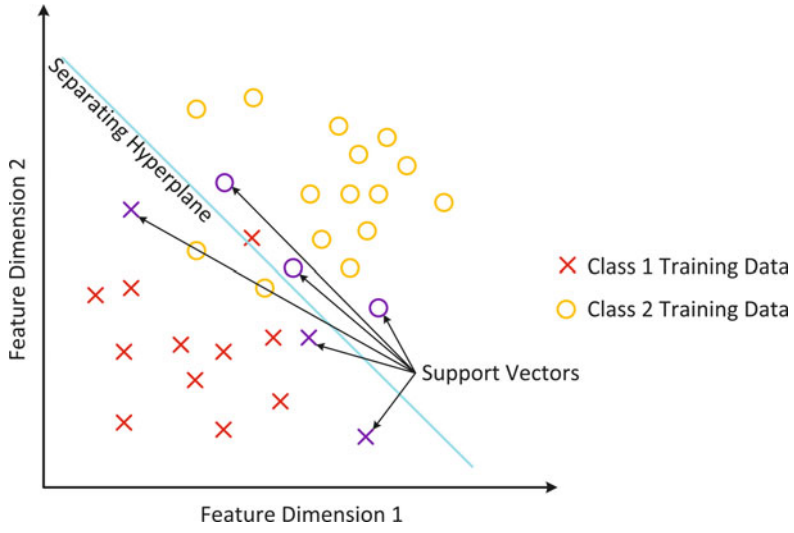

Fig. 7.7 Conceptual overview of linear support vector machines (SVM)

boundary between the two classes such that the separating hyperplane is defined as a linear combination of these support vectors. Classification (in the test phase) is then carried out by determining which side of the hyperplane the test vector lies on. Further, distance from the hyperplane may be used as an estimate of confidence of the decision. The distance, $d_{\mathbf{w}}(\overline{\mathbf{x}})$ of a vector $(\overline{\mathbf{x}})$ from the hyperplane, $\mathbf{w}$, may be estimated as:

$$d_{\mathbf{w}}(\overline{\mathbf{x}}) = \sum_{i=1}^{N_s} \alpha_i \widehat{\mathbf{x}}_{\mathbf{i}}^T \overline{\mathbf{x}} + \mathbf{b} \tag{7.1}$$

Where, $\widehat{\mathbf{x}}_{\mathbf{i}} \in \mathcal{R}^N$ are the i^{th} support vectors in the N dimensional feature space, N_s is the number of support vectors, $\mathbf{b} \in \mathcal{R}^N$ is the offset of the hyperplane from the origin, and α_i are the weights corresponding to the i^{th} support vector. A positive value of $d_{\mathbf{w}}$ corresponds to one class and a negative value to the other.

7.2.3.2 Emotion Models: Nonlinear Support Vector Machines

Nonlinear support vector machines are conceptually identical to linear SVMs with one difference. Instead of operating in the feature space, the features are mapped to a different space (typically of higher dimensionality) via a nonlinear transform and the best hyperplane on the transformed space is used for classification (see Fig. 7.8). The distance of a test vector from the hyperplane in the transformed space is given by

$$\widehat{d}_{\mathbf{w}}(\overline{\mathbf{x}}) = \sum_{i=1}^{N_s} \alpha_i \Phi\left(\widehat{\mathbf{x}}_{\mathbf{i}}\right)^T \Phi(\overline{\mathbf{x}}) + \mathbf{b}_{\Phi} \tag{7.2}$$

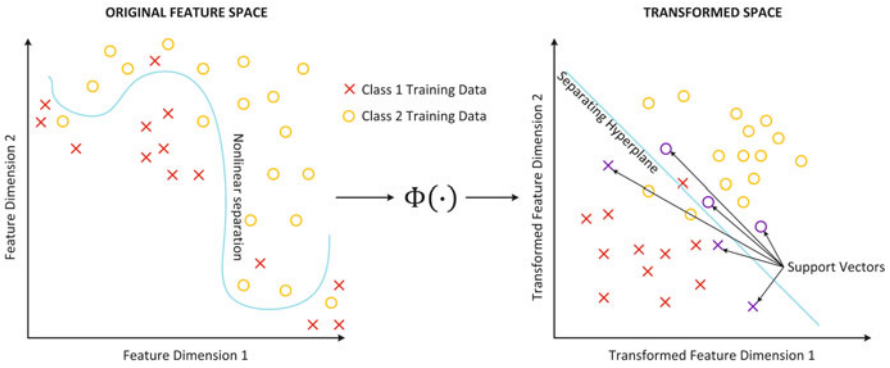

Fig. 7.8 Nonlinear SVM involves a nonlinear transformation of the original feature space followed by a linear separation in the transformed space which corresponds to a nonlinear boundary in the original space

Where, $\Phi : \mathcal{R}^N \rightarrow \mathcal{R}^M$ is the nonlinear transformation from the original N dimensional feature space to the M dimensional transformed space (typically, $M \gg N$).

The 'kernel trick' for nonlinear SVMs refers to the use of a kernel function that operates on pairs of vectors in the original space to give the inner product of the corresponding pair of vectors in the transformed space. i.e.,

$$\widehat{d}_{\mathbf{w}}(\overline{\mathbf{x}}) = \sum_{i=1}^{N_s} \alpha_i \mathrm{K}_{\Phi}\left(\widehat{\mathbf{x}}_{\mathbf{i}}, \overline{\mathbf{x}}\right) + \mathbf{b}_{\Phi} \tag{7.3}$$

Where, $\mathrm{K}_{\Phi} : \mathcal{R}^N \times \mathcal{R}^N \rightarrow \mathcal{R}$ is the kernel function corresponding to the transformation $\Phi(\cdot)$ such that $\mathrm{K}_{\Phi}\left(\widehat{\mathbf{x}}_{\mathbf{i}}, \overline{\mathbf{x}}\right) = \Phi\left(\widehat{\mathbf{x}}_{\mathbf{i}}\right)^T \Phi\left(\overline{\mathbf{x}}\right)$. This allows nonlinear support vector machines to be used without actually transforming the feature vectors into a high dimensional space since $\mathrm{K}_{\Phi}(\cdot,\cdot)$ operates directly on the original feature space. In addition, when the kernel function, K_{Φ} (which defines Φ or vice versa), is appropriately chosen the underlying transformed space maybe an infinite dimensional space.

Support vector machines are adept at handling high dimensional feature spaces, which makes them particularly suited to current emotion recognition systems that involve a front-end which estimates a high dimensional representation of utterances. Also, in general SVM training is formulated as a convex problem and consequently the training algorithm may be globally optimal [78]. However, a drawback to using SVMs is that there is no systematic approach to selecting the best kernel function. Often emotion recognition systems that make use of support vector machines choose the kernel function and its parameters heuristically. Some commonly chosen kernels include the linear kernel, the polynomial kernel, the radial basis kernel and the KL divergence kernel. It should be noted that the KL divergence kernel is a common choice in other fields of speech processing, particularly speaker verification. The use of linear support vector machines on GMM supervectors is equivalent to using a KL divergence kernel in the underlying feature space of the GMMs [79]. A comparison of the performance of some common kernels in an SVM based approach to emotion recognition using GMM supervectors can be found in [80]. Descriptions of emotion recognition systems that use SVM based classifiers can be found in [53, 68, 81], and comparisons between support vector machines and other classifiers can be found in [50, 82].

7.2.4 Alternative Emotion Modelling Methodologies

As previously mentioned, currently the most widespread approach to emotion recognition involves the estimation of short-term (frame level) features from each utterance. This set of feature vectors is then represented by a single high dimensional vector which is then used to make a decision. However, this approach is by no means the only one or even a single, rigorously defined system. A minor variation

involves the addition of other features which are extracted at time scales larger than a frame but smaller than an utterance, which are also incorporated into the high-dimensional representation. Some of these features that are commonly utilised in current systems are described in Sect. 7.2.4.1. An alternative approach focusses on dynamic modelling of short term features using an appropriate back-end, such as a hidden Markov model (HMM) instead of the static modelling approach taken by the high-dimensional approach. This approach is briefly outlined in Sect. 7.2.4.2. Finally, some systems directly model the distribution of the short term features using back-ends such as Gaussian mixture models (GMM) instead of abstracting utterance level information to a high dimensional representation. In place of this abstraction, these systems tend to use other methods to compensate for variability in short term features that arise due to reasons other than emotions. Some of these compensation methods may also be used in the other approaches and are briefly outlined in Sect. 7.3.

7.2.4.1 Supra-Frame Level Feature

In addition to the short-term (frame level) features outlined in Sect. 7.2.1, other features, estimated at time frames longer than a frame but shorter than an utterance, are also commonly utilised in emotion recognition systems. These may generally be categorised as follows:

Voice Quality Features

While there is no generally accepted definition of voice quality, the term has been used to refer to the auditory impressions of the listener that are not accounted for by measurable parameters. For instance, voice types such as hoarse, harsh and breathy are considered voice qualities. Some, but not all of these qualities have been associated with the shape of the glottal pulse [83]. Commonly used voice quality features include jitter and shimmer that capture variations in pitch and energy, noise-to-harmonics ratio (HNR) and autocorrelation features. Most of these features are estimated from short sequences of frame level measures.

Linguistic Features

A range of features that capture linguistic patterns which may be characteristic of emotions are collectively referred to as linguistic features. The most common among these include bag-of-words (BOW) features, which are vectors comprising of counts/frequencies of occurrence of words from a predefined vocabulary, bag-of-*N*-grams, which are similar to BOW, and parts-of-speech (POS) features, which are representations of frequencies of word classes/part of speech.

Non-linguistic Acoustic Events

Descriptions of non-linguistic acoustic events are comparatively rare compared to linguistic features. They include descriptions of occurrence and position of events such as disfluencies, breathing, laughter etc. within an utterance.

An overview of a range of short and long term features can be found in [50] and a comparative study between different types of features can be found in [57]. Similar features have also been utilised in other paralinguistic classification systems [84].

7.2.4.2 Dynamic Emotion Models

Speech signals are quasi-stationary (within short intervals of around 20 ms) and sequentially varying in time (over periods longer than 20 ms) in order to express information. Dynamic modelling of these sequential patterns is the corner stone of automatic speech recognition (ASR) systems and a possible approach to modelling emotional content. In ASR, hidden Markov models (HMMs) are the most commonly utilised dynamic models of feature sequences due to their generative stochastic framework and the existence of a range of techniques that may be employed to train, adapt and modify them. Hidden Markov models have also been employed in a number of emotion recognition systems [64, 66, 68, 85, 86], although they are not used as widely as the static modelling approach involving high-dimensional representations of utterances. Some comparative studies have suggested that dynamic modelling via HMMs are less suited to the task of emotion recognition than the static modelling approach outlined previously (Fig. 7.4) in some contexts [40, 87].

A hidden Markov model (HMM) is a doubly stochastic model with an underlying stochastic process that is not directly observable (hidden), but is linked through another set of stochastic processes that produces an observable sequence of symbols. In the context of pattern classification, a sequence of features (observable symbols) is modelled as being generated by a sequence of states (the number of possible states is finite and unrelated to the number of possible observable symbols) which is not directly observable. At every time instant (corresponding to each of the features in the sequence), the model enters a new state (which may be the same state as the previous one) based on a transition probability distribution which depends on the previous state (Markovian property) and generates the observation (feature) at that instant based on a probability distribution that is associated with that state (regardless of when and how the state is entered). The possible observations (single- or multi-dimensional) may belong to a discrete (and finite) or a continuous set, and thus giving rise to discrete and continuous HMMs respectively.

Any HMM is characterised by the state transition probability distribution, the initial state distribution, and the state observation probability distributions. The state observation pdfs in a continuous HMM are usually modelled by Gaussian mixture models (GMMs). The problem of estimating the parameters of a HMM is a difficult one and does not have an analytical solution. Typically iterative procedures, such as the Baum–Welch method, are used. An overview of hidden Markov models

including the Baum–Welch method to estimate the models can be found in [88]. Apart from HMMs, other dynamic modelling approaches include recurrent neural networks (RNN) [89], which are standard feed forward neural networks with additional feedback connections. The feedback connections allow for the existence of some sort of memory in the system and allow for past inputs in influence decisions made about the present input. A drawback of recurrent neural networks is that the magnitude of the influence of past inputs may decay or increase exponentially over time and the long short-term memory recurrent neural networks (LSTM-RNN) [90] were introduced to overcome this problem [91]. LSTM-RNN have also been used as dynamic models in speech based emotion recognition systems [41, 56]. LSTM-RNNs have also been used to build systems that tackle the regression problem with dimensionally labelling of emotions [35].

7.3 Dealing with Variability

Perfect features that are completely representative of emotions (or any other class of interest in any speech based problem) and no other factors are non-existent. Consequently emotion recognition systems may either attempt to choose features that exhibit low levels of variability due to other factors or they may compensate (implicitly or explicitly) for this variability. It is reasonable to assume that two major sources of variability that would affect an emotion recognition system are phonetic variability (variability in features reflecting what is being said) and speaker variability (variability in features reflecting characteristics of the speaker) and these are the focus of this section. These additional sources of variability in turn affect the 'classification rules' inferred by the back-end and degrade classification performance [92–94].

Given that feature spaces in emotion recognition systems are almost universally of a larger dimensionality than three, variability in the feature space cannot be directly visualised. However, some sort of visualisation could be useful to illustrate the ideas discussed in this section and hence the t-SNE algorithm [95] is employed to map feature spaces onto a 2-dimensional plane and scatter plots of the mapped points are shown. The t-SNE algorithm aims to preserve the local structure of the data and perhaps illuminate some of the global structure such as the presence of clusters. It does so by mapping Euclidean distances between data points in both the high (original feature space) and low (mapped 2-dimensional space) dimensional spaces to probability distributions and matching them [95].

7.3.1 Phonetic Variability in Emotion Recognition Systems

The durations of phonemes are typically in the range of tens of milliseconds to a few hundreds of milliseconds. Most speech recognition systems model these phonemes with a three (or five) state hidden Markov model, each state of which is designed to

be representative of a (quasi)stationary sequence of feature vectors. Consequently phonetic variability would manifest as short term variability. i.e., variations in short sequences of feature vectors. The duration of emotions are however significantly longer, spanning multiple words or utterances. From the point of view of emotion recognition systems aiming to discriminate between the longer term properties of features corresponding to emotions, the effect of phonetic variability would be akin to noise in the short term feature space. While the most obvious approach to dealing with phonetic variability is to explicitly compensate for it, most current emotion recognition systems deal with it implicitly. Specifically, the commonly undertaken approach of obtaining a high dimensional representation using functionals (Sect. 7.2.2.1) of the set of short term feature vectors from an utterance imparts a level of robustness against phonetic variability. This can be explained by the fact that the statistical descriptors that constitute the high-dimensional representation capture global (utterance level) characteristics while being relatively unaffected by local variations (across a few frames corresponding to phonetic durations). Scatter plots of frame level features (Fig. 7.9a) and utterance level high dimensional representations obtained using functionals (Fig. 7.9b) projected onto 2-dimensions (using the t-SNE algorithm) are depicted here to illustrate this idea. Both the frame level and utterance

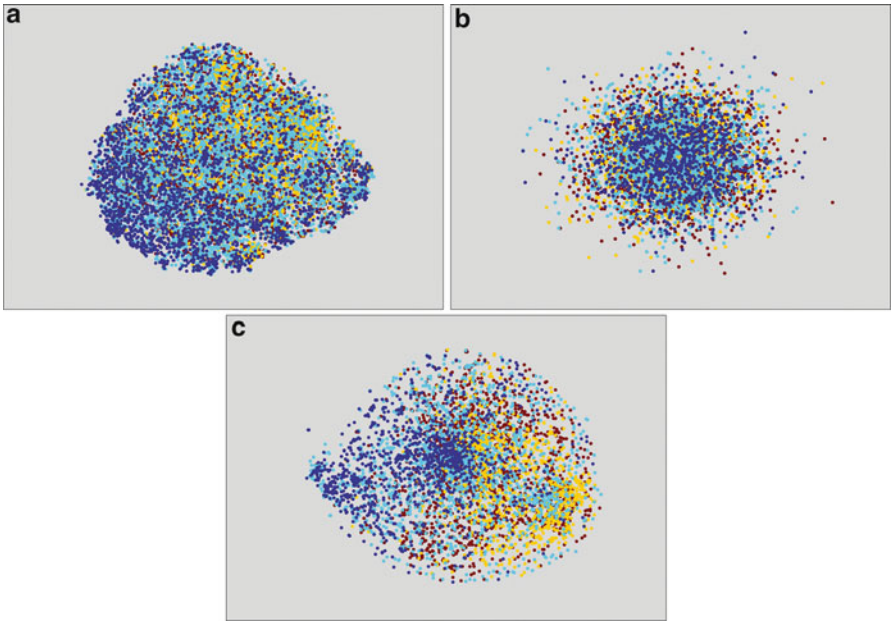

Fig. 7.9 Scatter plots of vectorial representations projected onto 2-dimensions using t-SNE algorithm (**a**) frame level MFCC feature; (**b**) utterance level high dimensional vector of functionals; (**c**) utterance level GMM supervectors (Data from IEMOCAP—blue corresponds to 'Anger', cyan to 'Happiness/Excited', yellow to 'Sadness' and maroon to 'Neutral')—The high dimensional utterance level representations are more robust to phonetic variability and exhibit tighter emotion clusters

level representations were extracted from speech data from the IEMOCAP database [96], containing four emotions from ten different speakers.

Using a GMM (Gaussian mixture model) supervector to represent an utterance (Sect. 7.2.2.2) would also be similarly robust since the GMM approximates the probability distribution of the features, which is insensitive to frame-to-frame variations when reliably estimated over a sufficient number of frames. Further, if suitably detailed spectral/cepstral features (such as MFCCs) are used, a GMM estimated on sufficient speech can serve as a universal background model (UBM), whose structure captures the acoustic/phonetic landscape. This UBM can then be adapted (MAP adaptation) to match the statistical properties of any target speech. Emotion recognition systems that use these adapted GMMs, either through the use of supervectors or by directly using them as emotion models, operate on the differences between the models including phoneme-specific differences. Consequently they are less affected by phonetic variability that is common across all of emotions. This robustness to phonetic variability also underpins the GMM-UBM approach to speaker recognition. Figure 7.9c shows a scatterplot of GMM supervector representations of speech from the IEMOCAP database [96] projected onto 2-dimensions using the t-SNE algorithm.

7.3.2 Speaker Variability

Speaker characteristics change very gradually over a period of years or decades. Hence, from the point of view of an emotion recognition system, the influence of speaker characteristics on speech features can for all intents and purposes be considered a complex but constant effect over the duration of an utterance and across all utterances from the same speaker. This is in contrast to phonetic variability, which manifests as characteristic variability over short sequences of frames.

As previously mentioned, most state-of-art emotion recognition systems implicitly cater to phonetic variability. However, this is not true of speaker variability and consequently speaker variability appears to be a more significant issue, that needs to be explicitly dealt with, in many commonly utilised features [97]. An attempt to quantify speaker variability in terms of speaker specific changes to emotion models (using GMMs) is reported in [61], with results suggesting that at least a part of the variability manifests as speaker specific shifts in feature vector clusters.

Approaches to compensate for speaker variability in emotion classification systems can be broadly categorised into those that explicitly personalise the systems towards a target speaker or those that alter the feature vectors or models of their distributions to minimise the effect of speaker variability on them (refer Fig. 7.10). The former category includes systems with back-ends trained exclusively on data from the target speaker [67], i.e., speaker dependent systems, and those with a generic back-end that is then suitably adapted towards target speakers [98, 99]. The latter category consists of techniques, referred to herein as speaker normalisation techniques, which aim to reduce speaker variability either in the feature domain or in the domain of models of feature distributions.

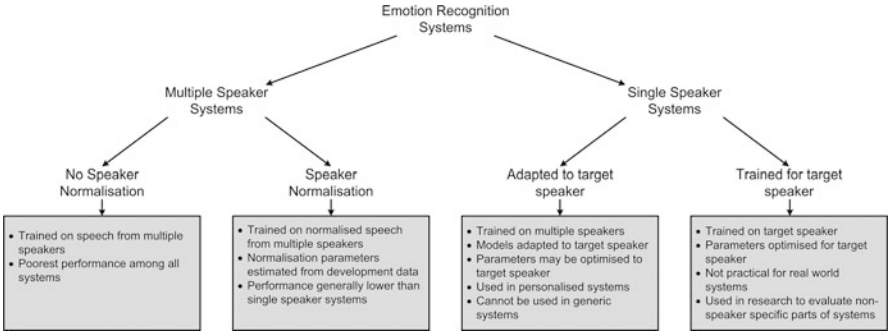

Fig. 7.10 Overview of the different approaches to dealing with speaker variability

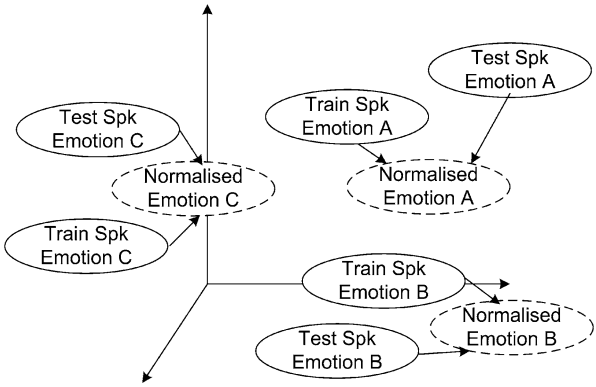

Fig. 7.11 Conceptual illustration of speaker normalisation

7.3.2.1 Speaker Normalisation

If the feature vectors corresponding to different emotions can be thought of as occupying different regions of the feature space (with the amount of overlap being proportional to the confusability between the overlapping emotions), the distribution of these regions is speaker-specific to some degree. Therefore models trained on data from one (or more) speaker(s) may not coincide with the regions corresponding to another and hence result in lower classification accuracy. Speaker normalisation can then be thought of as an attempt to address this issue by modifying the feature vectors for each speaker in a manner such that the emotion regions for different speakers align in the modified feature space (dashed ellipses in Fig. 7.11).

Feature normalisation methods may operate either in the feature domain or in the model domain. Both aim to minimise the effect of speaker variability on the statistical properties of the feature vector distributions. Specifically, feature domain techniques modify feature vectors directly [100–102] while model domain techniques modify the representation of models (such as supervectors) [103, 104].

7.3.2.2 Speaker Adaptation

Speaker adaptation techniques are motivated by the inability to perfectly separate effects of speaker variability from that of emotion variability. This observation is supported by the observation that speaker independent emotion classification systems do not perform as well as speaker dependent ones even if they incorporate speaker normalisation. Rather than normalise the features in order to minimise the mismatch between trained models and the test speaker, an alternative approach is to adapt the emotion-specific models of the back-end towards a target speaker. This is potentially superior to normalisation since it does not remove any information from the feature space. It is also perhaps the only approach that models both phonetic and speaker information (for e.g., using GMMs of sufficient complexity to model phonetic information followed by the use of MAP adaptation to create speaker specific models). An adaptation approach can adapt initial emotion models estimated from training speakers' data to match the target speaker. Compared to the speaker normalisation approach conceptualised in Fig. 7.11, speaker adaptation can be thought of as attempting to modify initial models to match the regions of the target speaker (Fig. 7.12).

Two (related) key drawbacks of the speaker adaptation approach when compared with speaker normalisation are that (a) some speech from the target speaker is required for adaptation prior to the system being used and; (b) the approach is only applicable if there is one target speaker for a system (that is initially trained on a large dataset and adapted to the target speaker), or if there is a small pool of known target speakers and the identity of the speaker is known at the time of operation. Descriptions of speaker adaptation approaches proposed for emotion recognition can be found in [98, 104].

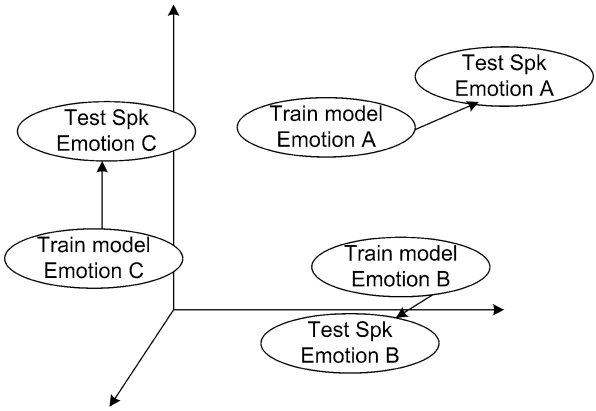

Fig. 7.12 Conceptual illustration of speaker adaptation approach

7.4 Comparing Systems

Unlike in some fields of speech processing, such as speaker verification and language identification, there is no standardised system of evaluation paradigms that allow emotion recognition systems to be directly compared. This is further exacerbated by the fact that the term emotion recognition can refer to more than one type of problem as previously mentioned (refer Sect. 7.1.3). While there has been some effort to address this situation in the form of emotion challenges [70, 71], these are still in their infancy. Further, even when different publications report recognition accuracies evaluated on the same database with the same metric, they may still not be directly comparable if they define the training and test sets differently. However over the years some databases have been utilised more than others and in many cases researchers use consistent definitions of training and test sets. Also, the existence of a large variety of databases has certain advantages as well. Specifically, different databases exhibit different characteristics allowing a wider range of issues to be explored than would be possible if all researchers worked on a single database [40, 105, 106]. This diversity in characteristics among database also motivates the use of multiple databases in serious investigations of emotion recognition systems. Overviews of commonly used databases can be found in [50, 107]. Finally, over the past 5 years there have been emotion recognition challenges [70, 71] which invited researchers to submit systems evaluated in an identical manner to allow direct comparisons, and these have helped address some of the above issues.

The submissions to the INTERSPEECH 2009 challenge are collectively one of the largest groups of directly comparable systems that classify emotions according to basic categorical labels, and some of them are listed in Table 7.1 to highlight some of the different approaches taken by emotion recognition systems. The 5-class classification accuracies of the systems are reported in terms of unweighted average recall (UAR) which was the metric used in the challenge, and is perhaps currently the most widely used metric for quantifying the accuracy of categorical emotion classification systems. The unweighted average recall (UAR) is defined as follows:

$$UAR = \frac{1}{\mathcal{C}} \sum_{i=1}^{\mathcal{C}} \frac{\eta_i}{N_i} \tag{7.4}$$

Where, $\mathcal{C}$ is the number of classes (emotions), N_i is the total number of test samples corresponding to emotion i and η_i is the number of test samples corresponding to emotion i that were classified correctly. When the test set is balanced, i.e., when there are an equal number of test samples from each class, the UAR is equivalent to the weighted average recall (WAR) which is the ratio of the total number of correctly classified test samples to the total number of test samples. The advantage of using the UAR over the WAR is that when the test set is not balanced, UAR values are not skewed by classification performance within a dominant class (a class with a large number of test samples compared to other classes). It weights each class accuracy equally, or in other words, class accuracies are not weighted by the number of test samples when averaged.

Table 7.1 Comparison of system approaches taken to INTERSPEECH 2009 emotion challenge (List and UAR taken from [94])

	System highlights	Five-class UAR (%)
Kockmann et.al. [60]	– MFCC + SDC + feature normalisation – GMM supervector – JFA channel (speaker) variability compensation	41.7
Bozkurt et. al. [108]	– Spectral + cepstral features + deltas – HMM posterior probabilities as features – GMM emotion models – Linear fusion of GMM likelihoods	41.6
Lee et. al. [109]	– Hierarchical binary tree classification structure – Bayesian logistic regression/SVM classifiers – High dimensional feature representation with functionals – Z-norm feature normalisation	41.6
Vlasenko and Wendemuth [110]	– MFCC + deltas + delta–deltas – VTLN + CMS feature normalisation – HMM based models	41.4
Luengo et. al. [111]	– Spectral features + GMM subsystem – Prosodic features + SVM subsystem – High dimensional feature representation with functionals – SVM based fusion	41.4
Planet et. al. [112]	– Spectral + voice quality features – High dimensional feature representation with functionals – Feature selection – SMO and naïve Bayes classifiers	41.2
Dumouchel et. al. [113]	– MFCC + deltas + delta–deltas – GMM emotion models	39.4
Vogt and André [114]	– Cepstral + spectral + prosodic + voice quality features – High dimensional feature representation with functionals – Feature selection – Naïve Bayes classification	39.4
Barra Chicote et. al. [115]	– MFCC + pitch + energy + deltas + delta–deltas – Dynamic Bayesian network based classification	38.2

An interesting observation that can be made from Table 7.1 is that a number of very different system approaches produced similar results. This is in contrast to other fields of speech processing such as speech recognition, speaker verification and language identification, all which have just a few well established dominant approaches. It should be noted that UAR as a performance metric does not shed any light on how accurate a system is at detecting/classifying individual emotions. Confusion matrices are much more suited for this purpose. However, not every

system submitted to the challenge reported a confusion matrix and general trends were hard to discern from the ones that were reported. A follow up emotion recognition challenge, to classify emotions according to categorical labels, on a different dataset was held as part of Interspeech-2013 [71] and descriptions of systems submitted to this challenge can be found in [116–119].

Given the current state of research into emotion recognition, it is advisable when conducting experiments on novel emotion recognition systems to (1) include a well-known feature set in the experimental work, even if only as a point of reference with which to compare a new feature set; (2) include a well-known classifier in the experimental work, even if only as a point of reference with which to compare a newly proposed classifier; (3) analyse at least two databases, at least one of which should preferably be well known in the literature and easily available to other researchers; (4) clearly explain how the various database partitions have been used; (5) use at least one well-known evaluation metric (e.g. UAR); and (6) during discussion, make every effort to try to compare new results with previously published results.

7.5 Conclusions

This chapter has presented an overview of current approaches to speech based emotion recognition systems. Specifically, rather than provide descriptions of complete systems, the chapter describes common techniques and outlines where they are commonly used in emotion recognition systems. This approach was taken due to the lack of any single established dominant approach and is instead an attempt to identify critical concepts and techniques common to many of current approaches.

Given the lack of a universally accepted 'theory of emotions', together with the expense and subjectivity of emotional database annotation, it is not surprising that the field of speech based emotion recognition is not as mature as other fields of speech based research such as speech and speaker recognition. This state of affairs also means that systems that are considered state-of-the-art today may be completely revamped in the future. Finally, while the identification of the optimal approach to emotion recognition is the ultimate goal of the field, a number of intermediate goals can be identified which would lead to this ultimate goal. Some research challenges that are most closely associated with the ideas discussed in this chapter are:

- There are many different types of emotion recognition applications and even approaches (various categorical and ordinal classification and regression configurations). There is no definitive formulation of the 'emotion recognition problem', although categorical classification between the 'big five' emotions is currently popular.
- Emotion recognition systems in general implicitly or explicitly account for both phonetic and speaker variability. Even if the most promising approaches to emotion recognition seem superficially simple (e.g. brute-forced functionals of features with tuned SVM), there is still research to be conducted to better explain

the means by which these account for variability and why they are successful. Understanding this better might be expected to lead to further improvements in system design.
- There is a fairly wide gap between speaker independent and speaker dependent classification performance and it is not clear how this gap may be bridged.
- Perhaps more than any other major speech classification application, there is an ongoing problem of comparability between published papers, which are mostly disjoint with respect to the features and/or classifiers and/or data sets that are investigated. Community recognition of the need for 'reference' features, classifiers and databases is still emerging.
- In the absence of a single, suitably large and varied 'default' emotion recognition database, system evaluation on multiple databases and cross-database testing offers the promise of robustness against overfitting due to aggressive tuning of hyperparameters.

A few other contemporary challenges and observations that are relevant to the field but not directly related to the concepts outlined above include:

- It is not clear what emotion recognition systems will or should look like as the amount of data tends towards infinity. Unlike speech recognition or speaker recognition, collecting but especially labelling emotion data is expensive and not definitive. However, annotation of emotional data on a massive scale can be expected at some point in the coming years.
- Emotion recognition is a stepping stone to a huge array of mental state/speaker trait/speaker attitude/speaker intention classification and regression problems (whose solutions have many similarities with emotion recognisers), many of which are only just beginning to be investigated. Some may well eclipse emotion recognition in terms of commercial importance.
- What is the optimal timescale over which emotions should be annotated? Is it possible to estimate likely points of emotion onset, offset and transition? If so, how should this be done?
- Attempts to 'break' emotion recognition systems, at least as a learning exercise should be encouraged. For instance, how badly does a system perform with even slightly different parameter choices and why?, how does performance vary across different speakers, how bad does it get?, etc.
- What can be learnt about characteristics of emotions from simpler, prototypical datasets such as neutral vs anger? (while avoiding the dangers inherent in focusing too much on small sets of strong prototypical emotions).
- What, if any, measures of confidence of decision may be obtained and how can they be used in applications?
- Why is recognition/regression along the valence axis harder than along the arousal axis and what is the best approach to deal with this?

In addition to these, some further speculative challenges include:

- Can more complex states, which combine more than one emotion or mental state, be recognised, and if so, how? For example, speaker intent might involve

more than one emotion (similar to the 'overlapping emotion' issue). Also, in some cases a complex state may comprise a sequence of emotions, e.g. surprise followed by happiness, or frustration alternating with anger. This might be analogous to the relationship between a phoneme recogniser and a speech recogniser.

- How quickly can emotions be accurately recognised? Successful human interaction often depends on an extremely rapid assessment of the mood of the conversational partner. Can this be done?
- Can psychological, psychiatric and/or emotional intelligence measures be estimated using emotion recognition style systems? For example, can they be used to assess an individual's impulse control or working memory or characteristics such as dominance, defensiveness, outbursts, etc. in conversation?
- If extremely rich data are available for certain individuals that allow for estimation of very accurate speaker-dependent models, can these be leveraged to obtain more accurate recognition for an unknown (newly presenting) speaker? It is very likely that large scale emotion annotation might occur for only very small numbers of people either as a research or a commercial reality.
- What devices will incorporate emotion recognition systems in the future, and hence what additional modalities (for e.g., facial expression, eye activity, physiological responses, muscle tension, touch, etc.) should be investigated more closely in terms of their connection with emotional speech? Further, how may these modalities be optimally incorporated into emotion recognition systems?
- Can the emotion of a group of people be recognised? Can the emotional dynamics of the group be modelled?

References

1. A.R. Damasio, A second chance for emotion, in *Cognitive Neuroscience of Emotion*, ed. by R. Lane et al. (Oxford University Press, New York, 2000), pp. 12–23
2. K. Scherer, On the nature and function of emotion: a component process approach, in *Approaches to Emotion*, ed. by K.R. Scherer, P. Ekman (Lawrence Erlbaum Associates, Inc., Hillsdale, 1984), pp. 293–317
3. S. Tompkins, *Affect Imagery Consciousness-Volume I the Positive Affects: The Positive Affects* (Springer Publishing Company, New York, 1962)
4. O. Mowrer, *Learning Theory and Behavior* (Wiley, New York, 1960)
5. G. Bower, Mood and memory. Am. Psychol. **36**, 129–148 (1981)
6. K.R. Scherer, What are emotions? And how can they be measured? Soc. Sci. Inf. **44**, 695–729 (2005)
7. P. Verduyn, I. Van Mechelen, F. Tuerlinckx, The relation between event processing and the duration of emotional experience. Emotion **11**, 20 (2011)
8. P. Verduyn, E. Delvaux, H. Van Coillie, F. Tuerlinckx, I. Van Mechelen, Predicting the duration of emotional experience: two experience sampling studies. Emotion **9**, 83 (2009)
9. A. Moors, P.C. Ellsworth, K.R. Scherer, N.H. Frijda, Appraisal theories of emotion: state of the art and future development. Emot. Rev. **5**, 119–124 (2013)
10. I. Fónagy, Emotions, voice and music. Res. Aspects Singing **33**, 51–79 (1981)
11. J. Ohala, Cross-language use of pitch: an ethological view. Phonetica **40**, 1 (1983)

12. E. Kramer, Judgment of personal characteristics and emotions from nonverbal properties of speech. Psychol. Bull. **60**, 408 (1963)
13. I.R. Murray, J.L. Arnott, Toward the simulation of emotion in synthetic speech: a review of the literature on human vocal emotion. J. Acoust. Soc. Am. **93**, 1097–1108 (1993)
14. I. Pollack, H. Rubenstein, A. Horowitz, Communication of verbal modes of expression. Lang. Speech **3**, 121–130 (1960)
15. K.R. Scherer, R. Banse, H.G. Wallbott, Emotion inferences from vocal expression correlate across languages and cultures. J. Cross Cult. Psychol. **32**, 76–92 (2001). doi:10.1177/0022022101032001009
16. C. Darwin, *The Expressions of Emotions in Man and Animals* (John Murray, London, 1872)
17. P. Ekman, An argument for basic emotions. Cogn. Emot. **6**, 169–200 (1992)
18. B. De Gelder, Recognizing emotions by ear and by eye, in *Cognitive Neuroscience of Emotion*, ed. by R. Lane et al. (Oxford University Press, New York, 2000), pp. 84–105
19. T. Johnstone, K. Scherer, Vocal communication of emotion, in *Handbook of Emotions*, ed. by M. Lewis, J. Haviland, 2nd edn. (Guilford, New York, 2000), pp. 220–235
20. K.R. Scherer, Vocal communication of emotion: a review of research paradigms. Speech Comm. **40**, 227–256 (2003)
21. K. Scherer, Vocal affect expression: a review and a model for future research. Psychol. Bull. **99**, 143–165 (1986)
22. R. Frick, Communicating emotion: the role of prosodic features. Psychol. Bull. **97**, 412–429 (1985)
23. R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, J.G. Taylor, Emotion recognition in human–computer interaction. Signal Proc. Mag. **18**, 32–80 (2001)
24. R. Cowie, R.R. Cornelius, Describing the emotional states that are expressed in speech. Speech Comm. **40**, 5–32 (2003)
25. J. Averill, A semantic atlas of emotional concepts. JSAS Cat. Sel. Doc. Psychol. **5**, 330 (1975)
26. R. Plutchik, *The Psychology and Biology of Emotion* (HarperCollins College Div, New York, 1994)
27. R. Cowie, E. Douglas-cowie, B. Apolloni, J. Taylor, A. Romano, W. Fellenz, What a neural net needs to know about emotion words, in *Computational Intelligence and Applications*, ed. by N. Mastorakis (Word Scientific Engineering Society, Singapore, 1999), pp. 109–114
28. H. Scholsberg, A scale for the judgment of facial expressions. J. Exp. Psychol. **29**, 497 (1941)
29. H. Schlosberg, Three dimensions of emotion. Psychol. Rev. **61**, 81 (1954)
30. J.A. Russell, A. Mehrabian, Evidence for a three-factor theory of emotions. J. Res. Pers. **11**, 273–294 (1977)
31. R. Lazarus, *Emotion and Adaptation* (Oxford University Press, New York, 1991)
32. M. Schröder, Speech and emotion research: an overview of research frameworks and a dimensional approach to emotional speech synthesis (Ph. D thesis), Saarland University (2004)
33. P. Ekman, E.R. Sorenson, W.V. Friesen, Pan-cultural elements in facial displays of emotion. Science **164**, 86–88 (1969). doi:10.1126/science.164.3875.86
34. R.R. Cornelius, *The Science of Emotion: Research and Tradition in the Psychology of Emotions* (Prentice-Hall, Inc, Upper Saddle River, 1996)
35. M. Wöllmer, F. Eyben, S. Reiter, B. Schuller, C. Cox, E. Douglas-Cowie, R. Cowie, Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies, in *INTERSPEECH* (2008), pp. 597–600
36. M. Grimm, K. Kroschel, Emotion estimation in speech using a 3d emotion space concept, in *Robust Speech Recognition and Understanding*, ed. by M. Grimm, K. Kroschel (I-Tech, Vienna, 2007), pp. 281–300
37. H.P. Espinosa, C.A.R. García, L.V. Pineda, Features selection for primitives estimation on emotional speech, in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (2010), pp. 5138–5141
38. H. Gunes, B. Schuller, M. Pantic, R. Cowie, Emotion representation, analysis and synthesis in continuous space: A survey, in *2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)* (2011), pp. 827–834

39. T. Sobol-Shikler, Automatic inference of complex affective states. Comput. Speech Lang. **25**(1), 45–62 (2011). http://dx.doi.org/10.1016/j.csl.2009.12.005
40. B. Schuller, B. Vlasenko, F. Eyben, G. Rigoll, A. Wendemuth, Acoustic emotion recognition: A benchmark comparison of performances, in *ASRU 2009. IEEE Workshop on Automatic Speech Recognition & Understanding, 2009* (2009), pp. 552–557
41. M. Wollmer, B. Schuller, F. Eyben, G. Rigoll, Combining long short-term memory and dynamic Bayesian networks for incremental emotion-sensitive artificial listening. IEEE J. Sel. Top. Signal Process. **4**, 867–881 (2010)
42. R. Barra, J.M. Montero, J. Macias-Guarasa, L.F. D'Haro, R. San-Segundo, R. Cordoba, Prosodic and segmental rubrics in emotion identification, in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings* (2006), pp. I–I
43. M. Borchert, A. Dusterhoft, Emotions in speech – Experiments with prosody and quality features in speech for use in categorical and dimensional emotion recognition environments, in *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05* (2005), pp. 147–151
44. M. Lugger, B. Yang, An incremental analysis of different feature groups in speaker independent emotion recognition, in *ICPhS* (2007)
45. M. Pantic, L.J.M. Rothkrantz, Toward an affect-sensitive multimodal human–computer interaction. Proc. IEEE **91**, 1370–1390 (2003)
46. D. Ververidis, C. Kotropoulos, Emotional speech recognition: resources, features, and methods. Speech Commun. **48**, 1162–1181 (2006)
47. L. Vidrascu, L. Devillers, Five emotion classes detection in real-world call center data: The use of various types of paralinguistic features, in *Proceedings of International Workshop on Paralinguistic Speech – 2007* (2007), pp. 11–16.
48. S. Yacoub, S. Simske, X. Lin, J. Burns, Recognition of emotions in interactive voice response systems, in *Eighth European Conference on Speech Communication and Technology* (2003), pp. 729–732
49. D. Bitouk, R. Verma, A. Nenkova, Class-level spectral features for emotion recognition. Speech Commun. **52**, 613–625 (2010)
50. M. El Ayadi, M.S. Kamel, F. Karray, Survey on speech emotion recognition: features, classification schemes, and databases. Pattern Recogn. **44**, 572–587 (2011). doi:10.1016/j.patcog.2010.09.020
51. C. Lee, S. Narayanan, R. Pieraccini, Combining acoustic and language information for emotion recognition, in *Seventh International Conference on Spoken Language Processing* (2002), pp. 873–876
52. B. Schuller, A. Batliner, S. Steidl, D. Seppi, Emotion recognition from speech: Putting ASR in the loop, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009* (2009), pp. 4585–4588
53. B. Schuller, G. Rigoll, M. Lang, Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture, in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04)*, vol. 1 (2004) pp. I-577–I-580
54. S. Mozziconacci, D. Hermes, Role of intonation patterns in conveying emotion in speech, in *14th International Conference of Phonetic Sciences* (1999), pp. 2001–2004
55. F. Burkhardt, W. Sendlmeier, Verification of acoustical correlates of emotional speech using formant-synthesis, in *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion* (2000), pp. 151–156
56. M. Wöllmer, F. Eyben, B. Schuller, E. Douglas-Cowie, R. Cowie, Data-driven clustering in emotional space for affect recognition using discriminatively trained LSTM networks, in *INTERSPEECH* (2009), pp. 1595–1598
57. A. Batliner, S. Steidl, B. Schuller, D. Seppi, T. Vogt, J. Wagner, L. Devillers, L. Vidrascu, V. Aharonson, L. Kessous, Whodunnit–searching for the most important feature types signalling emotion-related user states in speech. Comput. Speech Lang. **25**, 4–28 (2011)

58. T. Kinnunen, H. Li, An overview of text-independent speaker recognition: from features to supervectors. Speech Commun. **52**, 12–40 (2010)
59. E. Ambikairajah, H. Li, W. Liang, Y. Bo, V. Sethu, Language identification: a tutorial. IEEE Circuits Syst. Magazine **11**, 82–108 (2011)
60. M. Kockmann, L. Burget, J. Cernocky, Brno University of Technology System for Interspeech 2009 Emotion Challenge, in *INTERSPEECH-2009* (2009), pp. 348–351
61. V. Sethu, J. Epps, E. Ambikairajah, Speaker variability in speech based emotion models – analysis and normalisation, in *ICASSP* (2013)
62. C. Clavel, I. Vasilescu, L. Devillers, G. Richard, T. Ehrette, Fear-type emotion recognition for future audio-based surveillance systems. Speech Commun. **50**(6), 487–503 (2008). http://dx.doi.org/10.1016/j.specom.2008.03.012
63. V. Sethu, Automatic emotion recognition: an investigation of acoustic and prosodic parameters (PhD Thesis), The University of New South Wales, Sydney (2009)
64. V. Sethu, E. Ambikairajah, J. Epps, On the use of speech parameter contours for emotion recognition. EURASIP J. Audio Speech Music Process. **2013**, 1–14 (2013)
65. C. Busso, S. Lee, S.S. Narayanan, Using neutral speech models for emotional speech analysis, in *INTERSPEECH* (2007), pp. 2225–2228
66. B. Schuller, G. Rigoll, M. Lang, Hidden Markov model-based speech emotion recognition, in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, vol. 2 (2003), pp. II-1–II-4
67. V. Sethu, E. Ambikairajah, J. Epps, Group delay features for emotion detection, in *INTERSPEECH-2007* (2007), pp. 2273–2276
68. C.M. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, S. Narayanan, Emotion recognition based on phoneme classes, in *INTERSPEECH* (2004)
69. D. Küstner, R. Tato, T. Kemp, B. Meffert, Towards real life applications in emotion recognition, in *Affective Dialogue Systems*, ed. by E. André, L. Dybkaer, W. Minker, P. Heisterkamp (Springer, Berlin, 2004), pp. 25–35
70. B. Schuller, S. Steidl, A. Batliner, The INTERSPEECH 2009 emotion challenge, in *INTERSPEECH-2009*, Brighton (2009), pp. 312–315
71. B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, S. Kim, The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism, in *Interspeech*, Lyon (2013)
72. C.-C. Lee, E. Mower, C. Busso, S. Lee, S. Narayanan, Emotion recognition using a hierarchical binary decision tree approach. Speech Commun. **53**(11), 1162–1171 (2011). http://dx.doi.org/10.1016/j.specom.2011.06.004
73. D. Morrison, R. Wang, L.C. De Silva, Ensemble methods for spoken emotion recognition in call-centres. Speech Commun. **49**(2), 98–112 (2007). http://dx.doi.org/10.1016/j.specom.2006.11.004
74. J. Cichosz, K. Slot, Emotion recognition in speech signal using emotion-extracting binary decision trees, *Doctoral Consortium. ACII* (2007)
75. M.W. Bhatti, W. Yongjin, G. Ling, A neural network approach for human emotion recognition in speech, in *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04.*, vol. 2 (2004), pp. II-181–II-184
76. V. Petrushin, Emotion in speech: recognition and application to call centers, in *Conference on Artificial Neural Networks in Engineering* (1999), pp. 7–10
77. L. Chul Min, S.S. Narayanan, R. Pieraccini, Classifying emotions in human–machine spoken dialogs, in *2002 IEEE International Conference on Multimedia and Expo, 2002. ICME '02 Proceedings*, vol. 1 (2002), pp. 737–740
78. C.C. Burges, A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**, 121–167 (1998). doi:10.1023/A:1009715923555
79. W.M. Campbell, D.E. Sturim, D.A. Reynolds, A. Solomonoff, SVM based speaker verification using a GMM supervector kernel and NAP variability compensation, in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings* (2006), pp. I-I

80. H. Hao, X. Ming-Xing, W. Wei, GMM supervector based SVM with spectral features for speech emotion recognition, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007* (2007), pp. IV-413–IV-416
81. O. Kwon, K. Chan, J. Hao, T. Lee, Emotion recognition by speech signals, in *INTERSPEECH-2003* (2003), pp. 125–128
82. O. Pierre-Yves, The production and recognition of emotions in speech: features and algorithms. Int. J. Hum. Comput. Stud. **59**(7), 157–183 (2003). http://dx.doi.org/10.1016/S1071-5819(02)00141-6
83. D.G. Childers, C.K. Lee, Vocal quality factors: analysis, synthesis, and perception. J. Acoust. Soc. Am. **90**, 2394–2410 (1991)
84. B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, S. Narayanan, Paralinguistics in speech and language—state-of-the-art and the challenge. Comput. Speech Lang. **27**(1), 4–39 (2013). http://dx.doi.org/10.1016/j.csl.2012.02.005
85. T.L. Nwe, S.W. Foo, L.C. De Silva, Speech emotion recognition using hidden Markov models. Speech Commun. **41**, 603–623 (2003)
86. A. Nogueiras, A. Moreno, A. Bonafonte, J. Mariño, Speech emotion recognition using hidden Markov models, in *Proceedings of EUROSPEECH-2001* (2001), pp. 2679–2682
87. B. Vlasenko, B. Schuller, A. Wendemuth, G. Rigoll, Frame vs. turn-level: emotion recognition from speech considering static and dynamic processing, in *Affective Computing and Intelligent Interaction*, ed. by A. Paiva, R. Prada, R. Picard (Springer, Berlin, 2007), pp. 139–147
88. L. Rabiner, B. Juang, An introduction to hidden Markov models. IEEE ASSP Magazine **3**, 4–16 (1986)
89. P. Chang-Hyun, S. Kwee-Bo, Emotion recognition and acoustic analysis from speech signal, in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 4 (2003), pp. 2594–2598
90. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
91. A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, (Wiley, 2013)
92. A. Batliner, R. Huber, in *Speaker Characteristics and Emotion Classification*, ed. by C. Müller, vol. 4343 (Springer, Berlin, 2007), pp. 138–151
93. C. Busso, M. Bulut, S.S. Narayanan, in *Toward Effective Automatic Recognition Systems of Emotion in Speech*, ed. by J. Gratch, S. Marsella (Oxford University Press, New York, 2012)
94. B. Schuller, A. Batliner, S. Steidl, D. Seppi, Recognising realistic emotions and affect in speech: state of the art and lessons learnt from the first challenge. Speech Commun. **53**, 1062–1087 (2011). doi:10.1016/j.specom.2011.01.011
95. L. Van der Maaten, G. Hinton, Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 85 (2008)
96. C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, S. Narayanan, IEMOCAP: interactive emotional dyadic motion capture database. Lang. Res. Eval. **42**, 335–359 (2008)
97. V. Sethu, E. Ambikairajah, J. Epps, Phonetic and speaker variations in automatic emotion classification, in *INTERSPEECH-2008* (2008), pp. 617–620
98. D. Ni, V. Sethu, J. Epps, E. Ambikairajah, Speaker variability in emotion recognition – An adaptation based approach, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 5101–5104
99. K. Jae-Bok, P. Jeong-Sik, O. Yung-Hwan, On-line speaker adaptation based emotion recognition using incremental emotional information, in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2011), pp. 4948–4951
100. V. Sethu, E. Ambikairajah, J. Epps, Speaker normalisation for speech-based emotion detection, in *2007 15th International Conference on Digital Signal Processing* (2007), pp. 611–614
101. C. Busso, A. Metallinou, S.S. Narayanan, Iterative feature normalization for emotional speech detection, in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2011), pp. 5692–5695

102. B. Schuller, M. Wimmer, D. Arsic, T. Moosmayr, G. Rigoll, Detection of security related affect and behaviour in passenger transport, in *Interspeech*, Brisbane (2008), pp. 265–268
103. L. Ming, A. Metallinou, D. Bone, S. Narayanan, Speaker states recognition using latent factor analysis based Eigenchannel factor vector modeling, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 1937–1940
104. T. Rahman, C. Busso, A personalized emotion recognition system using an unsupervised feature adaptation scheme, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 5117–5120
105. M. Tahon, A. Delaborde, L. Devillers, Real-life emotion detection from speech in human–robot interaction: Experiments across diverse corpora with child and adult voices, in *INTERSPEECH* (2011), pp. 3121–3124
106. B. Schuller, B. Vlasenko, F. Eyben, M. Wollmer, A. Stuhlsatz, A. Wendemuth, G. Rigoll, Cross-corpus acoustic emotion recognition: variances and strategies. IEEE Trans. Affect. Comput. **1**, 119–131 (2010). doi:10.1109/T-AFFC.2010.8
107. E. Douglas-Cowie, N. Campbell, R. Cowie, P. Roach, Emotional speech: towards a new generation of databases. Speech Commun. **40**, 33–60 (2003)
108. E. Bozkurt, E. Erzin, C.E. Erdem, A.T. Erdem, Improving automatic emotion recognition from speech signals, in *INTERSPEECH* (2009), pp. 324–327
109. C.-C. Lee, E. Mower, C. Busso, S. Lee, S.S. Narayanan, Emotion recognition using a hierarchical binary decision tree approach, in *INTERSPEECH-2009* (2009), pp. 320–323
110. B. Vlasenko, A. Wendemuth, Processing affected speech within human machine interaction, in *INTERSPEECH* (2009), pp. 2039–2042
111. I. Luengo, E. Navas, I. Hernaez, Combining spectral and prosodic information for emotion recognition in the Interspeech 2009 emotion challenge, in *INTERSPEECH-2009* (2009), pp. 332–335
112. S. Planet, I. Iriondo, J. Socoró, C. Monzo, J. Adell, GTM-URL contribution to the INTERSPEECH 2009 emotion challenge, in *INTERSPEECH-2009* (2009), pp. 316–319
113. P. Dumouchel, N. Dehak, Y. Attabi, R. Dehak, N. Boufaden, Cepstral and long-term features for emotion recognition, in *INTERSPEECH-2009* (2009), pp. 344–347
114. T. Vogt, E. André, Exploring the benefits of discretization of acoustic features for speech emotion recognition, in *INTERSPEECH* (2009), pp. 328–331
115. R. Barra Chicote, F. Fernández Martínez, L. Lutfi, S. Binti, J.M. Lucas Cuesta, J. Macías Guarasa, J.M. Montero Martínez, R. San Segundo Hernández, J.M. Pardo Muñoz, Acoustic emotion recognition using dynamic Bayesian networks and multi-space distributions, in *INTERSPEECH 2009* (2009), pp. 336–339
116. O. Räsänen, J. Pohjalainen, Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech. Presented at the 14th Annual Conference of the International Speech Communication Association (Interspeech), Lyon (2013)
117. G. Gosztolya, R. Busa-Fekete, L. Tóth, Detecting autism, emotions and social signals using AdaBoost. Presented at the 14th Annual Conference of the International Speech Communication Association (Interspeech), Lyon
118. H.-y. Lee, T.-y. Hu, H. Jing, Y.-F. Chang, Y. Tsao, Y.-C. Kao, T.-L. Pao, Ensemble of machine learning and acoustic segment model techniques for speech emotion and autism spectrum disorders recognition. Presented at the 14th Annual Conference of the International Speech Communication Association (Interspeech), Lyon (2013)
119. V. Sethu, J. Epps, E. Ambikairajah, H. Li, GMM based speaker variability compensated system for Interspeech 2013 ComParE emotion challenge. Presented at the 14th Annual Conference of the International Speech Communication Association (Interspeech), Lyon (2013)

Chapter 8
Speaker Diarization: An Emerging Research

Trung Hieu Nguyen, Eng Siong Chng, and Haizhou Li

Abstract *Speaker diarization* is the task of determining "Who spoke when?", where the objective is to annotate a continuous audio recording with appropriate speaker labels corresponding to the time regions where they spoke. The labels are not necessarily the actual *speaker identities*, i.e. speaker identification, as long as the same labels are assigned to the regions uttered by the same speakers. These regions may overlap as multiple speakers could talk simultaneously. Speaker diarization is thus essentially the combination of two different processes: *segmentation*, in which the speaker turns are detected, and *unsupervised clustering*, in which segments of the same speakers are grouped. The clustering process is considered as unsupervised problem since there is no prior information about the number of speakers, their identities or acoustic conditions (Meignier et al., Comput Speech Lang 20(2–3):303–330, 2006; Zhou and Hansen, IEEE Trans Speech Audio Process 13(4):467–474, 2005). This chapter presents the fundamentals of speaker diarization and the most significant works over the recent years on this topic.

8.1 Overview

Figure 8.1 shows the typical components of a speaker diarization system. The *signal processing* module applies standard techniques such as: pre-emphasis, noise reduction and/or beamforming to improve the *signal-to-noise ratio* (SNR) and to reduce undesired noises. The *feature extraction* module transforms the raw audio signal into feature vectors in which speaker-related characteristics are captured and unintended properties such as noises are suppressed. Subsequently, only useful feature vectors are retained for further processing. These vectors are generally corresponding to speech frames and the selection of these frames is implemented in the *speech activity detection* (SAD) module. Finally, at the heart of a speaker diarization system is

T.H. Nguyen (✉) • H. Li
Institute for Infocomm Research, Singapore, Singapore
e-mail: thnguyen@i2r.a-star.edu.sg; hli@i2r.a-star.edu.sg

E.S. Chng
Nanyang Technological University, Singapore, Singapore
e-mail: aseschng@ntu.edu.sg

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_8

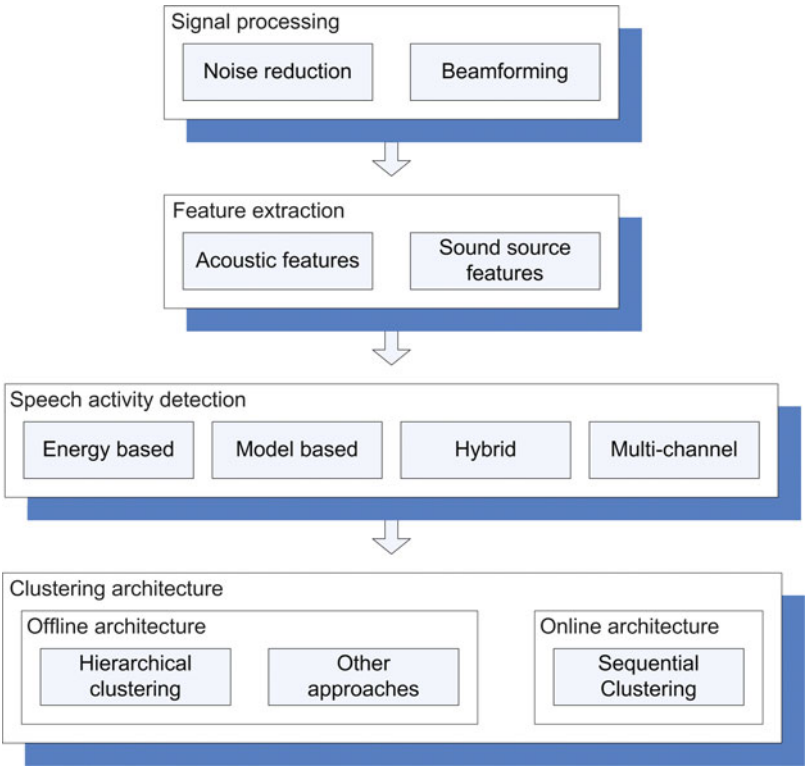

Fig. 8.1 Typical components of a speaker diarization system

the *clustering architecture* which defines the strategies and approaches to perform speaker clustering from the unlabeled feature vectors. These important components are covered thoroughly in the upcoming sections.

8.2 Signal Processing

Signal processing techniques are typically applied to raw signals to produce well calibrated signals suitable for specific tasks. Among such techniques, speech enhancement are commonly used in many speaker diarization systems to enhance the SNR and attenuate interferences. Depending on the availability of audio channels, single or multi-channel processing techniques could be employed for these purposes, and the common approaches are: (1) *Wiener filtering* and (2) *acoustic beamforming*.

8.2.1 Wiener Filtering

Given two processes: $s[k]$ the signal to be estimated, and $y[k]$ the observed noisy signal, which are *jointly wide-sense stationary*, with known covariance functions $r_s[k]$, $r_y[k]$, and $r_{sy}[k]$ respectively. A particular case is that of a signal corrupted by additive noise $n[k]$:

$$y[k] = s[k] + n[k] \tag{8.1}$$

The problem is to estimate the signal $s[k]$ as a function of $y[k]$. A widely adopted solution to this problem is *Wiener filtering* [150], which is used to produce an estimate of desired signal by linear time-invariant filtering an observed noisy signal, assuming known stationary signal and noise spectra, and additive noise. Wiener filtering gives the optimal way of filter out the noisy components, so as to give the best L^2-norm reconstruction of the original signal. Interested readers may refer to [150] for the solutions.

In the speaker diarization community for the NIST Rich Transcription evaluation [41–43], the Qualcomm-ICSI-OGI front end [6] is commonly used to perform Wiener filtering. Most state-of-the-art speaker diarization systems [47,101,127,153] applied Wiener filtering to all audio channels for speech enhancement before filtered and summed to produce a beamformed audio channel. In van Leeuwen and Konečný [79], the filtering, however, was applied after beamforming. The authors observed no difference in performance with the benefit of reduction in computational cost since only one channel was processed.

8.2.2 Acoustic Beamforming

In the scenarios where multiple audio channels are accessible, each one can have different characteristics, and the recorded audio quality therefore varies across the channels. For speaker diarization, one may select the best quality channel, for e.g. the highest *signal to noise ratio* (SNR), and work on this selected signal as traditional single channel diarization system. However, a more widely adopted approach is to perform *acoustic beamforming* on multiple audio channels to derive a single enhanced signal and proceed from there. The techniques for acoustic beamforming is a broad field of research on its own. Nevertheless, due to the nature of diarization task where a priori information such as microphone types and their positions are not given, robust and simple techniques are thus preferred. The favored approach, which are commonly used in many systems in NIST Rich Transcription Evaluation for meeting domains, is adaptive beamforming with filter-and-sum technique, a more general version of delay-and-sum beamforming. For clarity, the procedures for filter-and-sum beamforming technique are summarized here.

Given an array of M microphones, the filter-and-sum output z at instance k is defined as:

$$z[k] = \sum_{m=1}^{M} W_m\,[k]\; y_m\big[k + \Delta_m\,[k]\,\big] \tag{8.2}$$

where $y_m[k]$ is the signal for each channel and $\Delta_m[k]$ is the relative delay between each channel and the reference channel at instance k, $W_m[k]$ is the relative weight of microphone m at instance k with $\sum_{m=1}^{M} W_m[k] = 1$. The *time delay of arrival* (TDOA) $\Delta_m[k]$ is commonly estimated via cross-correlation methods such as *generalized cross correlation with phase transform* (GCC-PHAT) [69] as presented in Sect. 8.3.2.

For researchers who are not in the field of array processing, there are several freely available implementations of beamforming source code on the web and one such popular toolkit used by many researchers in the recent NIST 2009 Rich Transcription evaluation benchmark is known as BeamformIt [11]. The toolkit implemented an enhanced delay-and-sum algorithm, together with many multichannel techniques described in [91] which are rather relevant to speaker diarization research. The techniques include: automatic selection of reference microphone channel for GCC-PHAT computation, adaptive weighting of channel based on SNR or cross-correlation metric, and two-pass Viterbi decoding for smoothing spurious TDOA values. These techniques are applied to stabilize the TDOA values before the signals are beamformed.

8.3 Feature Extraction

Raw speech signal is normally converted into a sequence of feature vectors carrying characteristic information about the signal; this step is referred to as feature extraction. In the field of speaker diarization, as well as speaker recognition in general, the information that we want to retain is the speaker-related properties. Many types of features have been studied in the literature; commonly used features for speaker diarization could be broadly organized into two categories: (1) *acoustic features*, and (2) *sound-source features*.

8.3.1 Acoustic Features

Speech is produced when air is forced from the lungs through the vocal cords and along the vocal tract. Different speech sounds are generated by varying the shape of the *vocal tract* and its *mode of excitation*. The variations occur relatively slowly in the order of 20 ms. Thus, for short frames of 20–30 ms, speech signal can be

considered to be *quasi-stationary* and the *short-term features* are extracted to model the shape of the vocal tract or the excitation or the combination characteristics of both.

8.3.1.1 Short-Term Spectral Features

Short-term spectral features are based on the *spectral envelope*, the shape of the discrete Fourier transform (DFT) magnitude spectrum, of a short time *windowed frame* of speech (typically 20–30 ms). The effectiveness of these features are based on the observations that:

- The phonetic segments in speech appears as energy fluctuation over time in different frequency bands. This is useful for representing the phonetic contents in speech recognition.
- The spectral envelope contains information about the resonance properties of the vocal tract which depends on both phonetics and speakers. This is the most informative part of the spectrum in speaker recognition.

Popular spectral features are Mel Frequency Cepstral Coefficients (MFCC) [31], Linear Prediction Cepstral Coefficients (LPCC) [85] and Perceptual Linear Prediction Cepstral (PLPC) Coefficients [56]. These features differ mainly in the *analysis of time-frequency* and in the techniques for *frequency smoothing*.

In frequency analysis, an important notion is *critical-band* [10] which refers to the capability of human auditory system to localize and process information within the frequency range. The detection of signals within this frequency range is not sensitive to and less affected by interference signals outside of this critical bandwidth. This bandwidth is non-linearly dependent on frequency and a number of functions approximating these critical bandwidths were proposed, among which, two popular functions are *Bark scale* [164] and *Mel scale* [126]. MFCC filter bank follows Mel frequency scale, whereas in PLPC, the spectrum is filtered by a trapezoidal-shaped filter bank with Bark frequency scale.

Another differential factor among various spectral features is the frequency smoothing techniques being used. The frequency smoothing techniques are generally applied to enhance and preserve the *formant* information, which are the pattern of resonances and can be observed from the spectral envelope. To capture this pattern, the spectral envelope in MFCC features is derived from the FFT *power spectrum*, while in LPC and PLPC, the spectrum is approximated by a linear predictor with *all-pole model*.

For MFCC, LPCC and PLPC feature extraction, in the final step the spectral representation is transformed to *cepstral coefficients*, in which the coefficients are nearly *orthogonal*. This property is desirable as it is beneficial for modeling purpose and leads to significant reduction in the number of parameters to be estimated. Particularly, when using *Gaussian* or *Gaussian mixtures* model, diagonal covariance matrices with *uncorrelated* components could be used instead of full covariance matrices.

Given these alternative features, however, MFCCs, sometimes with their first and/or second derivatives, are more widely adopted in the community of speaker diarization research. In contrast to speech recognition, higher order cepstral coefficients are retained since they capture more speaker-specific information, yet there is no consensus on the order of MFCCs. Typically, 16–20 coefficients are used in most of the current state-of-the-art diarization systems [45, 47, 119, 146]. Nonetheless, Ajmera and Wooters [9] reported the diarization results using both LPCC and MFCC features. They observed that LPCCs perform better during clean speech, while MFCCs work better in case of noisy conditions. In another attempt, Wooters et al. [152] compared the performance of MFCC and PLP features, with the empirical evidence that MFCCs slightly outperform PLPs.

Apart from these common spectral features, some lesser known features were also explored for speaker diarization task in the literature. The LIA system [92, 93] performed speaker segmentation using 20th order *linear cepstral features* (LFCC) augmented by the energy.

8.3.1.2 Prosodic Features

Prosodic features are supra-segmental, they are not confined to any one segment, but occur in some higher level of an utterance. Prosodic units are marked by phonetic cues including pause, pitch, stress, volume, and tempo. The most important prosodic feature is the *fundamental frequency* ($F0$). Other common prosodic features are: duration, speaking rate, and energy distribution/modulations [117]. Combining prosodic features with spectral features has been shown to be effective for speaker verification, especially in noisy condition. Recently, these features have been adopted in several speaker diarization systems and showed promising results. In El-Khoury et al. [38], a difference between the averages of the $F0$ between speech segments was calculated and used as merging criterion for bottom-up clustering. In Friedland et al. [48], the authors investigated the speaker discriminability of 70 long-term features, most of them prosodic features. They applied *Fisher Linear Discriminant Analysis* (LDA) to rank these 70 prosodic and long-term features by their speaker discriminative power. The authors showed improvement in speaker diarization results when combining the top-ten ranked prosodic and long-term features with regular MFCCs. In a recent paper, Imseng and Friedland [59] proposed the use of prosodic features to obtain initial clusters, which are crucial in many state-of-the-art agglomerative speaker diarization systems. The proposed approach achieved significant improvement over the baseline system.

8.3.2 Sound Source Features

When multiple microphone recordings are accessible, the relative *time delay of arrival* (TDOA) between the different microphones can be estimated. Assuming the

speakers are not changing position, those features can be used in speaker diarization [99]. It has been shown that TDOA improves the speaker diarization significantly in combination with conventional spectral features [100].

Given a pair of microphones i and j, let $x_i[k]$ and $x_j[k]$ be the windowed signals from microphone i and j respectively. The cross-correlation matrix between the two signals is defined as

$$\mathrm{R}_{x_i x_j}[\tau] = \mathbf{E}\left\{x_i[k] \cdot x_j^*[k-\tau]\right\} \tag{8.3}$$

where $\mathbf{E}\{.\}$ denotes the expectation. In practice, it is estimated as

$$\mathrm{R}_{x_i x_j}[\tau] = \frac{1}{2N} \sum_{k=-N}^{N} x_i[k] \cdot x_j^*[k-\tau] \tag{8.4}$$

where N is the length of the windowed signals (in terms of number of samples), for reliable estimation, the window size is typically at least 500 ms. It is generally assumed that the signals picked up by the two microphones i and j are similar with one being the delayed version of the other by a time Δ_{ij}. Δ_{ij} is then estimated by maximizing the cross-correlation function:

$$\Delta_{ij} = \underset{-N \le \tau \le N}{\arg\max}\, \mathrm{R}_{x_i x_j}[\tau] \tag{8.5}$$

In real applications, however, there are many external factors such as ambient noises, reverberation etc. that could affect the estimation of time delay and it is shown that cross-correlation is not robust against these issues. To address this problem, Knapp and Carter [69] introduced a general version named the *Generalized Cross-Correlation* (GCC), which is defined as:

$$\mathrm{R}_{x_i x_j}[\tau] = \mathbf{E}\left\{\left(h_i[k] * x_i[k]\right) \cdot \left(h_j[k-\tau] * x_j^*[k-\tau]\right)\right\} \tag{8.6}$$

where $h_i[k]$ and $h_j[k]$ are the filter coefficients. It is apparent from the GCC equation that it is simply the cross-correlation computed on the filtered signals. Generally, GCC for long windowed signals is computed in the frequency domain for efficiency. The *generalized cross power spectral density* (GXPSD) [69] can be expressed as:

$$\mathbf{\Phi}_{x_i x_j}[l] = \left[H_i[l] X_i[l]\right] \cdot \left[H_j[l] X_j[l]\right]^* \tag{8.7}$$

where X_i, X_j, H_i, H_j are the discrete Fourier transform of x_i, x_j, h_i, and h_j correspondingly, with l being the discrete frequency index. Rearranging the above equation

$$\mathbf{\Phi}_{x_i x_j}[l] = H_i[l] H_j^*[l] X_i[l] X_j^*[l] \tag{8.8}$$

$$= \mathbf{\Psi}_{ij} X_i[l] X_j^*[l] \tag{8.9}$$

where

$$\mathbf{\Psi}_{ij} = H_i[l]H_j^*[l]$$

being the weighting function. Various weighting functions have been studied in the literature including: *Roth filter* [112], *the smoothed coherence transform* (SCOT) [26], *the phase transform* (PHAT) [69], *the Eckart filter* [37], and *the Hannon and Thomson filter* [69]. For general applications, PHAT is widely adopted as it is shown to be robust against a wide range of conditions [69]. Its definition is:

$$\mathbf{\Psi}_{ij}^{\text{PHAT}}[l] = \frac{1}{|X_i[l]X_j^*[l]|} \tag{8.10}$$

In summary, the time delay Δ_{ij} between microphone i and j can be estimated as:

$$\mathbf{\Phi}_{x_i x_j}[l] = \frac{X_i[l]Xj^*[l]}{|X_i[l]X_j^*[l]|} \tag{8.11}$$

$$\mathrm{R}_{x_i x_j}[\tau] = \mathscr{F}^{-1}\{\mathbf{\Phi}_{x_i x_j}[l]\} \tag{8.12}$$

$$\Delta_{ij} = \underset{-N\leq\tau\leq N}{\arg\max}\, \mathrm{R}_{x_i x_j}[\tau] \tag{8.13}$$

with $\mathscr{F}^{-1}\{.\}$ denoting the inverse Fourier transform.

8.3.3 Feature Normalization Techniques

8.3.3.1 RASTA Filtering

RASTA filtering [57] is mainly applied to removes slow channel variations. It is equivalent to a band-pass filtering of each frequency channel through an IIR filter with the transfer function:

$$H(z) = 0.1 \times \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{z^{-4} \times (1 - 0.98z^{-1})} \tag{8.14}$$

Filtering could be performed in either log spectral or cepstral domain. In the cepstral domain, the low and high cut-off frequency define the frequency range in which the cepstral change within this range is preserved. RASTA is applied on the MFCC features before estimating speaker models in the MIT Lincoln Laboratory diarization systems [110].

8.3.3.2 Cepstral Mean Normalization

Cepstral Mean Normalization (CMN) is typically employed to minimize the effect of session variations, which occur with the change of channel characteristics. It is calculated by first estimating the cepstral mean across an utterance or a window of N frames and then subtracting the mean from each cepstral vector to obtain the normalized vector. As a result, the long-term average of any observation sequence (the first moment) is zero. When the audio stream is processed online, a dynamic CMN approach is applied, where the cepstral mean $\boldsymbol{\mu}_k$ at time k is updated as follows:

$$\boldsymbol{\mu}_k = \alpha \mathbf{C}[k] + (1 - \alpha)\boldsymbol{\mu}_{k-1} \tag{8.15}$$

where α is a time constant (typically, around 0.001), $\mathbf{C}[k]$ is the cepstral vector at time k and $\boldsymbol{\mu}_{k-1}$ is the dynamic cepstral mean at time $(k-1)$. In Reynolds and Torres-Carrasquillo [110], MFCC features for each cluster is processed with CMN to increase robustness against channel distortion in their offline speaker diarization systems. While in Zamalloa et al. [158], the dynamic CMN approach is applied in their online speaker tracking system.

8.3.3.3 Feature Warping

In order to avoid the influence of background noises and other non-speaker related events, *feature warping* is proposed to condition and conform the individual feature streams such that they follow a specific target distribution over a window of speech frames. Normally, the target distribution is chosen to be following Gaussian shape [102]. In the context of speaker diarization, Sinha et al. [123] and Zhu et al. [161] apply this normalization technique for each short segment using a sliding window of 3 s in the clustering stage.

8.4 Speech Activity Detection

Speech activity detection (SAD) identifies audio regions containing speech from any of the speakers present in the recording. Depending on the domain of the data being used, the non-speech regions may contain silence, laughing, music, room noise, or background noise. The use of a speech/non-speech detector is an important part of speaker diarization system. The inclusion of non-speech frames into the clustering process makes it difficult to correctly differentiate between two speaker models. SAD could be broadly classified into four categories: (1) *energy-based speech detection*, (2) *model based speech detection*, (3) *hybrid speech detection*, and (4) *multi-channel speech detection*.

8.4.1 Energy-Based Speech Detection

Many energy-based speech detectors are proposed in the literature, however, with the diverse environments of audio recordings, the non-speech can be from a variety of noise sources, like paper shuffling, coughing, laughing, etc. energy-based methods have shown to be relatively ineffective in speaker diarization task [60,138]. Nevertheless, with its simplicity and speed, this approach has been adopted in several systems. In [27], Cassidy defines a threshold based on *root mean square* (RMS) and *zero crossing rate* (ZCR) of the audio signal to separate speech and silence.

8.4.2 Model Based Speech Detection

With the limitation of energy-based approach, in general, *model based* speech/non-speech detectors are frequently used in many speaker diarization systems as they are able to characterize various acoustic phenomena. The simplest system uses just two models for speech and non-speech such as in Wooters et al. [152]. A more complex system is described in Nguyen et al. [97] with four speech models including gender/bandwidth combinations. Noise and music are explicitly modeled in Gauvain et al. [51], and Zhu et al. [162]; the systems comprise of five classes: speech, music, noise, speech + music, and speech + noise. The speech + music and speech + noise models are used to help minimize the false rejection of speech occurring in the presence of music or noise, and this data is subsequently reclassified as speech [51, 54, 123, 162]. The classes can be broken down further, as in Liu and Kubala [81], there are five models for non-speech (music, laughter, breath, lip-smack, and silence) and three for speech (vowels and nasals, fricatives, and obstruents). In Meignier et al. [90], the acoustic segmentation system are designed in a hierarchical approach to provide finer classification. First, speech/non-speech is detected then the speech class is further classified as clean speech, speech with music, and telephone speech. Each category is subsequently separated by gender and two additional models representing female and male speech recorded under degraded conditions are then included to refine the final segmentation.

8.4.3 Hybrid Speech Detection

The model-based approach, however, has its own limitation: its models need to be trained with pre-labeled data using training set. This requires the data to be annotated with class labels and this process takes much effort. Moreover, depending on the complexity of the models, there might not be enough data to build these models. The performance of these models on unseen data (which in statistical

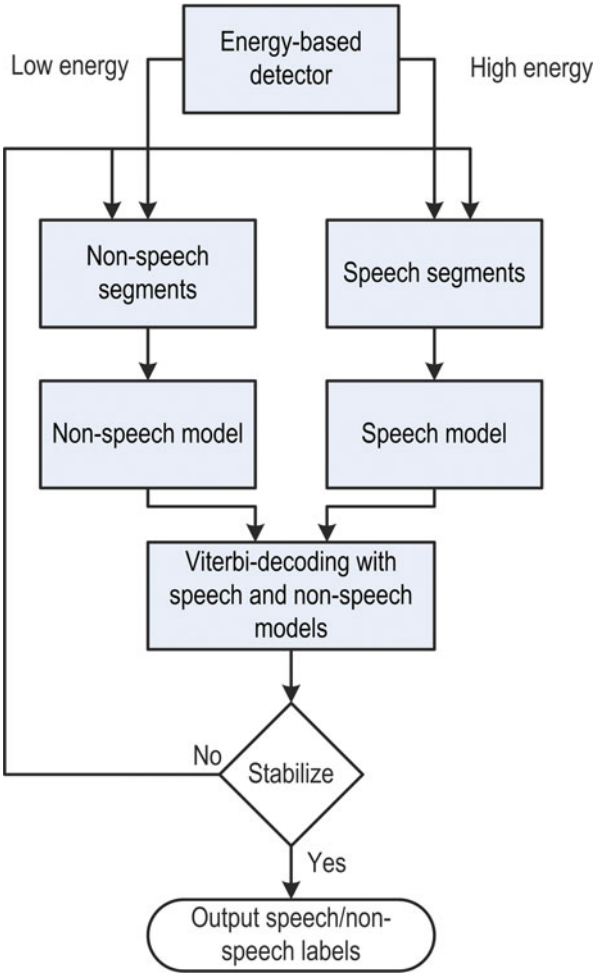

Fig. 8.2 Hybrid energy-based and model-based speech activity detector

machine learning is known as the term generalization) is also an important issue especially in the case where testing data is substantially different from development data. To mitigate these problems, the *hybrid approach* is proposed. This approach comprises of two stages: the first stage is a simple energy-based detector, the second stage is a model-based detector in which the models are trained on the test data itself, hence no training data is required. Figure 8.2 depicts a typical structure of a hybrid energy-based and model-based speech activity detector. In Anguera et al. [12, 16], the speech/non-speech regions were first detected by an enhanced energy-based SAD with low pass filtering in conjunction with duration constraint using *finite state machine* (FSM). From these labels, a *Hidden Markov*

model (HMM) were constructed with two states: a single Gaussian for modeling silence, and a *Gaussian Mixture Model* (GMM) for modeling speech. The detector then iteratively classifies the audio frames and re-trains both models until the overall likelihood converges. The proposed approach performs sufficiently well, however, in a more diverse environments, using energy-based detector to obtain initial labels may place some restrictions on the system as it is not possible to detect high energy noises. An improved speech detector is then suggested in Huijbregts et al. [58], and Wooters et al. [153] which are able to detect distinct non-speech segments. In this system, speech, silence or non-speech sounds regions are first detected by pre-trained models on broadcast news. Those regions with high confidence scores are then split to three classes: speech, non-speech with low energy, non-speech with high energy and high zero crossing rate. Three models are built up iteratively, and the audio is re-segmented a number of times.

8.4.4 Multi-Channel Speech Detection

In recent years, with the increasing availability of multi-channel audio, there have been a number of related efforts toward *multi-speaker speech activity detection*. In Wrigley et al. [154, 155], the authors performed a systematic analysis of features for classifying multi-channel audio into four sub-classes: local channel speech, crosstalk speech, local channel and crosstalk speech, and non-speech. They looked at the frame-level classification accuracy for each class with various features selected for analysis. A key result from this work is that, from among the 20 features examined, the single best performing feature for each class is one derived from cross-channel correlation. This result evidences the importance of cross-channel information for multi-channel detection task. Pfau et al. [104] proposed an *ergodic HMM* (eHMM) speech activity detector and as a post-processing step, the authors identified and removed crosstalk speech segment by thresholding cross-channel correlations which yields 12 % relative *frame error rate* (FER) reduction. In [74], Laskowski et al., proposed a scheme using a cross-channel correlation speech/non-speech detection. This scheme was later used in a multi-channel speech activity detection system that models vocal interaction between meeting participants with joint multi-participant models [73, 75, 76].

8.5 Clustering Architecture

Speaker clustering seeks to group all audio frames, segments from the same speakers together [159]. Ideally, this process produces one cluster for each speaker with all segments of a given speaker assigned to a single cluster. Different diarization

systems adopt different strategies for speaker clustering. However, in a broad sense, the clustering architectures fall into one of these categories: (1) *offline architecture*, or (2) *online architecture*. In offline architecture, all the feature vectors are observable by the system at all times and the algorithm could optimize through multiple iterations with no constraint on the execution time. While in online architecture, the features are presented to the system only when the data is available, the algorithm has no knowledge about the future and generally there is constraint on the latency, the time difference between when the result is obtained and when the data is available. Figure 8.3 illustrates the *hierarchical clustering* approach, which is the most popular clustering approach for offline speaker diarization systems. The following sections then discuss the components of various clustering architectures, with focus on offline speaker diarization systems and well-established approaches. Online speaker clustering architectures are discussed separately in Sect. 8.5.5 as not all mentioned techniques are suitable for real-time processing and special considerations are called for.

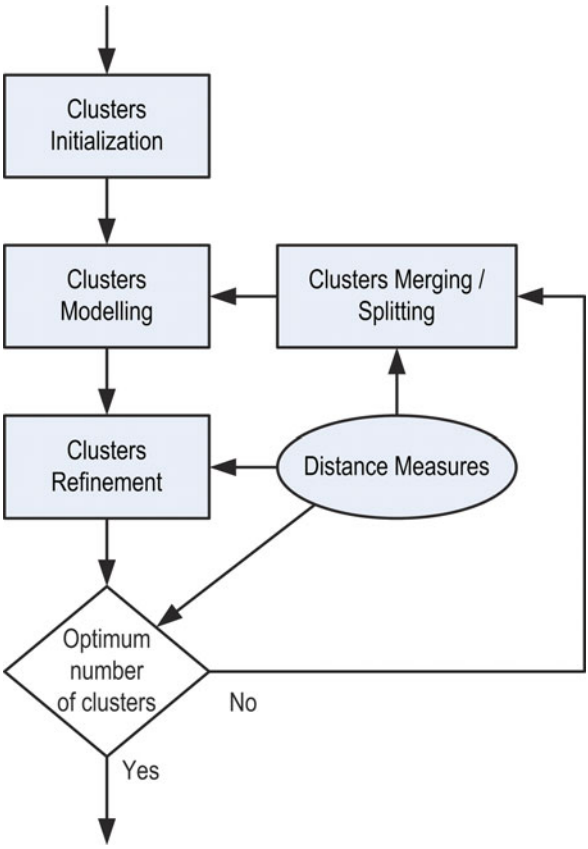

Fig. 8.3 Hierarchical clustering architecture

8.5.1 Speaker Modeling

At the heart of many speaker diarization systems is the choice of speaker modeling technique. As diarization is a part of speaker recognition, many modeling techniques in speaker verification and identification are also applicable. In this section, however, we only include those which have been adopted and shown to be effective for diarization tasks.

8.5.1.1 Gaussian Mixture Model

Since *Gaussian Mixture Model* (GMM) was initially introduced in the context of speaker modeling by Reynolds et al. [109], it has become the standard reference method in speaker recognition. A GMM is a probability distribution that is a convex combination of several *Gaussian distributions*. The mixture density is:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{x}) \tag{8.16}$$

where

- K is the number of mixtures.
- α_k is the prior probability of mixture k such that $\sum_{k=1}^{K} \alpha_k = 1$
- $f_k(\mathbf{x})$ is the component density of Gaussian distribution parametrized by mean $\boldsymbol{\mu}_k$ and covariance Σ_k:

$$f_k(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(\frac{-(\mathbf{x} - \boldsymbol{\mu}_k)^t \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)}{2} \right) \tag{8.17}$$

  with d being the dimension of the feature vector and $|\Sigma_k|$ being the determinant of Σ_k.

Given a sequence of observation vectors, the parameters of a GMM can be trained via the *Expectation Maximization* (EM) algorithm [35] to maximize the likelihood of the data. In speech processing, it is generally assumed that the observations in sequence $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are *independent and identically distributed* (i.i.d.). Accordingly, the likelihood of a GMM parametrized by Θ given observations sequence $\mathbf{X}$ is computed as:

$$p(\mathbf{X}|\Theta) = \prod_{i=1}^{i=N} p(\mathbf{x}_i|\Theta) \tag{8.18}$$

In practice, many systems restrict the covariance matrices of the GMM to be diagonal, since it is computationally expensive and requires more training data to estimate the parameters of a full-covariance GMM.

The choice of K, the number of mixtures, is highly empirical as there is no consistency among different systems [23, 79, 127, 153]. In the ISCI agglomerative speaker diarization systems [9, 18, 152], the authors suggested the use of variable complexities. The systems were initialized with a fixed number of Gaussian mixtures for all cluster models. Upon merging any two clusters, the new cluster model is generated with the complexity as the sum of both parent's Gaussian mixtures. Later, in [15], Anguera et al. proposed the concept of *cluster complexity ratio* (CCR), which assumes that the number of mixtures is linearly proportional to the number of features in the clusters, to initialize this parameter. Based on the similar assumption, in Leeuwen and Konečný [79], the *constant seconds per Gaussian* (CSPG) was used to determine the number of mixtures.

Mono-gaussian model uses a single Gaussian component with either full or diagonal covariance matrix as speaker model. Modeling with mono-gaussian is computationally efficient since only a small number of parameters need to be estimated. Although the accuracy is clearly behind GMM, it is sometimes the model of choice due to: lack of training data, or limitation on computational resource. In many speaker segmentation systems [14, 86], since the speaker segments are relatively short, mono-gaussian models with full covariance matrices were employed to detect speaker change points. In others [92, 93], diagonal covariance matrices were used. Reynolds and Torres-Carrasquillo [110] performed bottom-up clustering with BIC metric and mono-gaussian models with full covariance.

8.5.1.2 Hidden Markov Model

The *Hidden Markov Model* (HMM) [105] is a generative probabilistic model comprising of a finite number internal hidden states and these states are not visible to observer. Each hidden state is associated with an emission probability distribution and an observation can be generated according to this distribution. In speech processing, Gaussian or mixture of Gaussians are commonly used to model the emission probabilities. The transitions among hidden states are assumed to follow the first-order Markov process, they are specified by a transition probability matrix and an initial state distributions.

Formally, a HMM is completely specified by $\{\Pi, \mathbf{A}, \Theta\}$ with:

- A set of parameters of emission distribution conditioned on the hidden states: $\Theta = \{\Theta_1, \ldots, \Theta_N\}$ with N being the number of states.
- A matrix of transition probabilities

$$\mathbf{A} = \begin{pmatrix} a_{11} & \ldots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \ddots & a_{NN} \end{pmatrix} \tag{8.19}$$

  where a_{ij} is the transition probability from state i to state j.
- The initial state distributions $\Pi = \{\pi_1, \ldots, \pi_N\}$

In this chapter, $\Lambda = \{\Pi, \mathbf{A}, \Theta\}$ is denoted as the parameters of the HMM. When used for speech, the HMM usually has a left-to-right topology. Given a sequence of observation vectors $\mathbf{X}$, the parameters of the HMM are trained using EM algorithm to maximize the likelihood:

$$\Lambda^* = \arg\max_{\Lambda} p\left(\mathbf{X}|\Lambda\right) \tag{8.20}$$

The best hidden state sequence q_{best} is derived using the Viterbi algorithm [148], i.e.:

$$q_{\text{best}} = \arg\max_{q} p\left(\mathbf{X}, q|\Lambda\right) = \arg\max_{q} p\left(\mathbf{X}|q, \Lambda\right) \cdot p\left(q|\Lambda\right) \tag{8.21}$$

The likelihood of an observation vector $\mathbf{x}_n$ given state q_k, $p(\mathbf{x}_n|q_k)$, is generally modeled by a GMM.

The HMM-based speaker clustering framework was first presented by Ajmera et al. in [7]. Since then, it has been widely adopted in most state-of-the-art speaker diarization systems [23, 79, 153]. The LIA speaker diarization system [92, 93] also used HMM with different topology for speaker modeling. In their system, there is no duration constraint, each state of the HMM characterizes a speaker and the transitions model the speaker turns. On the other hand, Kim et al. [67] performed re-segmentation by applying a HMM-based classifier on segments of 1.5 s each with the assumption that no speaker change within each segment.

8.5.1.3 Total Factor Vector

With the success of the *total variability* approach in the task of speaker verification [32], it has been recently adapted to the problem of speaker diarization [118]. In this modeling technique, a speaker utterance is represented by a *supervector* $\mathbf{M}$ that consists of components from the *total variability subspace*, contains the speaker and channel variabilities simultaneously.

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w} + \boldsymbol{\varepsilon} \tag{8.22}$$

where $\mathbf{M}$ is a speaker and session dependent supervector, a supervector in this context is a stacked mean vectors from a GMM [25], and $\mathbf{m}$ is the speaker and session independent supervector commonly adapted from the *Universal Background Model* (UBM) supervector. In the speaker recognition terminology, the UBM is a large GMM (in terms of 512–2048 mixtures), trained on speech from many speakers (several hundred to several thousand), to represent the speaker independent distribution of acoustic features [108]. Matrix $\mathbf{T}$ is the rectangular matrix of low rank which contains the eigenvectors with the largest eigenvalues of the total variability covariance matrix, $\mathbf{w}$ is a low-dimensional random vector having a standard normal

distribution $\mathcal{N}(0, I)$, and the residual noise term $\boldsymbol{\varepsilon}$ covers the variabilities not captured by $\mathbf{T}$ [66]. The vector $\mathbf{w}$, with dimension in order of hundreds compared to the dimension in order of thousands of a supervector, is referred to as a *total factor vector*, an *identity vector* or an *i-vector*. In short, this modeling technique can be seen as a simple factor analysis for projecting a speech utterance from high-dimensional supervector space to the low-dimensional total variability space. Once projected to low-dimensional space, the applicability of many machine learning algorithms are then more straight forward.

Probably, the first attempt to make use of i-vector in the context of speaker diarization was presented by Shum et al. in [118]. In this paper, good diarization results on summed-channel telephone data with two speakers were reported with various dimensions of the i-vector from 40 to 600, in conjunction with *Principal Component Analysis* (PCA) for further dimension reduction and cosine distance metric for scoring. In the later work also by Shum et al. [120], the authors applied i-vector in the framework of spectral clustering [96] to extend the solution to diarization of telephone data with unknown number of participating speakers. In [122], with the motivation that the estimation of i-vectors for short segments is not reliable and may harm the clustering process especially at early phases, Silovsky and Prazak employed the two-stage clustering approach, using i-vector in the second stage, while the first stage using GMM for speaker modeling. They reported performance improvement over the standalone i-vector system. In [114], Rouvier and Meignier re-defined the speaker clustering as a problem of *Integer Linear Programming* (ILP) based on i-vectors and conclude that i-vector models are more robust than GMMs.

8.5.1.4 Other Modeling Approaches

8.5.1.4.1 Supervector

Supervector often refers to combining many low dimensional vectors into a higher dimensional vector. In speaker recognition terminology, supervector typically refers to Gaussian supervector [25], formed by stacking all the mean vectors of an adapted GMM. Supervector is widely used in many speaker verification systems together with *support vector machine* (SVM) classifier. These combinations have been shown to be effective and robust in many situations, probably due to the ability to capture the speech utterance statistics of supervector as well as the generalization capability of SVM in high dimensional space. In Tang et al. [128], supervector was used with either Euclidean or cosine distance metric to measure the distance among different speakers. These distances are then used to learn the speaker-discriminative acoustic feature transformation and the discriminative speaker subspace. They reported that the speaker clustering methods based on the GMM mean supervector and vector-based distance metrics outperform traditional methods based on statistical model and statistical model-based distance metrics.

8.5.1.4.2 Eigen Vector Space Model

Eigen vector space model (EVSM) [133] is inspired from the eigenvoice approach [71]. Each cluster is first modeled by a supervector, then all the super vectors are projected to a lower subspace by applying *Principal Component Analysis* (PCA) to obtain new supervectors with reduced dimension. These newly obtained vectors are termed eigen vector space models. It has been shown experimentally in [133] that clustering with EVSM and cosine distance metric consistently yielded higher cluster purity and lower Rand Index than the GLR-based method. In a later work, EVSM was also used in El-Khoury et al. [38] for cluster modeling in their hierarchical bottom-up clustering system.

8.5.2 Distance Measures

Many speaker diarization systems employ some kinds of distance metrics in one way or the other. In agglomerative systems, they are used to decide which clusters to merge and when to stop the clustering process. While in speaker segmentation, distance metrics are often used in conjunction with sliding windows to detect the speaker change points. On the other hand, these metrics also find some applications in cluster refinement and purification.

Many distance measures were proposed in the past and they can be broadly classified into two categories: (1) *template-based*, and (2) *likelihood-based*. The first kind compares the parameters of the models which are applied to the data. These are generally very fast to compute and often used as initial estimation or in real-time systems. The representatives of this kind are: *Symmetric Kullback-Leibler distance* (KL2) [121], *Divergence Shape Distance* (DSD) [24] and *Arithmetic Harmonic Sphericity* (AHS) [21]. The second group of distances require the evaluation of the fitness (likelihood) of the data given the representing models. These distances are slower to compute since the likelihood-score need to be evaluated for each data point, however their performance is better than those in the first category. In this chapter, the metrics in the second group are referred to as likelihood-based techniques, which are also the main focus of this section. Among them, the more popular distances are: the *Generalized Likelihood Ratio* (GLR) [151], the *Bayesian Information Criterion* (BIC) [29], the *Cross Likelihood Ratio* (CLR) [107] and the *Normalized Cross Likelihood Ratio* (NCLR) [77].

Let us consider two audio segments (i, j) with feature vectors $\mathbf{X}_i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_{N_i}^i\}$ and $\mathbf{X}_j = \{\mathbf{x}_1^j, \mathbf{x}_2^j, \ldots, \mathbf{x}_{N_j}^j\}$ respectively. For brevity, from now on we refer to the audio segments as $\mathbf{X}_i$ and $\mathbf{X}_j$. It is desirable that a proper distance metric would have a smaller value if these two segments belong to the same speaker and have a larger value if these two segments belong to different speakers.

8.5.2.1 Symmetric Kullback-Leibler Distance

Kullback-Leibler (KL) divergence between two random variables A and B is an information theoretic approach to measure the expected number of extra bits required to encode random variable A with a code that was designed for optimal encoding of B [30].

$$\mathrm{KL}(A||B) = \int_{-\infty}^{\infty} p_A(x) \log \frac{p_A(x)}{p_B(x)} dx \tag{8.23}$$

where p_A and p_B denote the pdf of A and B. *Symmetric Kullback-Leibler* (KL2) is simply a symmetric version of KL and is defined as:

$$\mathrm{KL2}(A, B) = \mathrm{KL}(A||B) + \mathrm{KL}(B||A) \tag{8.24}$$

When both A and B have Gaussian distributions, a closed form solution could be derived [24]:

$$\begin{aligned}\mathrm{KL2}(A, B) = &\frac{1}{2}\mathbf{tr}\left\{(\Sigma_A - \Sigma_B)\left(\Sigma_B^{-1} - \Sigma_A^{-1}\right)\right\} \\ &+ \frac{1}{2}\mathbf{tr}\left\{\left(\Sigma_A^{-1} + \Sigma_B^{-1}\right)(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T\right\}\end{aligned} \tag{8.25}$$

where Σ_A, Σ_B, $\boldsymbol{\mu}_A$, $\boldsymbol{\mu}_A$ are respectively the covariance matrices and means of p_A and p_B, and $\mathbf{tr}\{.\}$ denotes the trace of the matrix.

Given any two audio segments $\mathbf{X}_i$ and $\mathbf{X}_j$, they can be considered as random variables A and B and therefore the distance can be computed using the above formula.

KL2 is used in the work by Siegler et al. [121] to compute the distance between two sliding windows for speaker change point detection. It is also employed as cluster distance metric in several agglomerative speaker clustering systems [121, 160]. For their system, the authors [121] show that KL2 distance works better than the Mahalanobis distance.

When using KL2 as a distance metric for speaker clustering, the speaker models are often assumed to be Gaussian distributed since there is closed-form expression to compute KL2. This assumption may make the speaker models too simple to be able to capture the characteristics of individual speakers. There are some works in this direction to adapt KL2 metric to more complex models. In Ben et al. [20], a novel distance between GMMs was derived from the KL2 distance for the particular case where all the GMMs are mean adapted from a common GMM based on the principles of *Maximum A Posteriori* (MAP) adaptation. The speaker diarization results using this metric are shown to be better than BIC when the segmentation is of high quality. However, due to the sensitivity to segmentation errors, this metric

is not as robust as BIC in general. In Rogui et al. [113], the author proposed a divergence measurement method between GMMs which is based on KL divergence and allows fast computation of distance between GMMs.

8.5.2.2 Divergence Shape Distance

Observe that Eq. (8.25) comprises of two components in which one of them involves the mean vectors. As the mean vectors are easily biased by environment conditions, the *Divergence Shape Distance* (DSD) [156] is derived from the KL distance by eliminating this part. Therefore, the corresponding expression for DSD is:

$$\mathrm{DSD}(A, B) = \frac{1}{2}\mathbf{tr}\left\{(\Sigma_A - \Sigma_B)\left(\Sigma_B^{-1} - \Sigma_A^{-1}\right)\right\} \tag{8.26}$$

Kim et al. [67] employed DSD for speaker change detection; the authors showed that the DSD method is more accurate than the BIC approach in presence of short segments, while both approaches are equivalent on long segments.

8.5.2.3 Arithmetic Harmonic Sphericity

The *Arithmetic Harmonic Sphericity* (AHS) [21] assumes the distributions of random variables are Gaussian and it can be viewed as an arithmetic harmonic sphericity test on covariance matrices of pdfs of two random variables. The AHS is defined as:

$$\mathrm{AHS}(A, B) = \log\left(\mathbf{tr}\left\{\Sigma_A\Sigma_B^{-1}\right\} \times \mathbf{tr}\left\{\Sigma_B\Sigma_A^{-1}\right\}\right) - 2\log(d) \tag{8.27}$$

where d is the dimension of the feature vector.

8.5.2.4 Generalized Likelihood Ratio

Given two audio segments $\mathbf{X}_i$ and $\mathbf{X}_j$, let us consider the following hypothesis test:

- H_0: both segments are generated by the same speaker.
- H_1: each segment is from a different speaker.

The feature vectors of each speaker K are assumed to be distributed according to the generating *probability density function* g_K.

- Under hypothesis H_0: $\mathbf{X}_i \cup \mathbf{X}_j \sim g_{X_i X_j}$.
- Under hypothesis H_1: $\mathbf{X}_i \sim g_{X_i}$ and $\mathbf{X}_j \sim g_{X_j}$

Since the generating density functions g_{X_i}, g_{X_j}, and $g_{X_iX_j}$ are unknown, these functions are therefore required to be estimated from the observed data by means of *maximum likelihood* (ML) optimization. Denote f_{X_i}, f_{X_j}, and $f_{X_iX_j}$ respectively the ML estimated models of the generating densities g_{X_i}, g_{X_j}, and $g_{X_iX_j}$. The *Generalized Likelihood Ratio* (GLR) between two hypotheses is then defined by:

$$R = \frac{\mathscr{L}\left(\mathbf{X}_i \cup \mathbf{X}_j \,|\, f_{X_iX_j}\right)}{\mathscr{L}\left(\mathbf{X}_i \,|\, f_{X_i}\right)\mathscr{L}\left(\mathbf{X}_j \,|\, f_{X_j}\right)} \tag{8.28}$$

with $\mathscr{L}(\mathbf{X}|\, f_X)$ being the likelihood of the data $\mathbf{X}$ given the probability density function f_X. The feature vectors are assumed to be *independently and identically distributed* (i.i.d.), thus:

$$R = \frac{\prod_{k=1}^{N_i} f_{X_iX_j}\left(\mathbf{x}_k^i|\Theta_{f_{X_iX_j}}\right)\prod_{k=1}^{N_j} f_{X_iX_j}\left(\mathbf{x}_k^j|\Theta_{f_{X_iX_j}}\right)}{\prod_{k=1}^{N_i} f_{X_i}\left(\mathbf{x}_k^i|\Theta_{f_{X_i}}\right)\prod_{k=1}^{N_j} f_{X_j}\left(\mathbf{x}_k^j|\Theta_{f_{X_j}}\right)} \tag{8.29}$$

being $\Theta_{f_{X_i}}$, $\Theta_{f_{X_j}}$, and $\Theta_{f_{X_iX_j}}$ the parameter sets of the pdfs f_{X_i}, f_{X_j}, and $f_{X_iX_j}$ correspondingly. The distance d_{GLR} is the negative logarithm of the previous expression:

$$d_{\mathrm{GLR}} = -\log R \tag{8.30}$$

In Bonastre et al. [22], the GLR is used to segment the signal into speaker turns. In Adami et al. [5], an algorithm is specifically designed for two-speaker segmentation with GLR as distance metric. Another system for two-speaker segmentation is proposed by Gangadharaiah et al. [49], with GLR metric in the first segmentation step.

8.5.2.5 Bayesian Information Criterion

Bayesian Information Criterion (BIC) is a Bayesian approach to the model selection problem which is proposed by Schwarz [116]. The BIC value for a model M is defined as:

$$\mathrm{BIC}_M = \log \mathscr{L}(\mathbf{X}|M) - \lambda \frac{\#(M)}{2} \log N \tag{8.31}$$

where $\mathscr{L}(\mathbf{X}|M)$ denotes the likelihood of data $\mathbf{X}$ given model M, $\#(M)$ denotes the number of free parameters in M and N denotes the number of observations in X, λ is a tunable parameter dependent on the data. BIC is an approximation to the

posterior distribution on model classes. It is shown in [116] that maximizing BIC value also results in maximizing the expected value of the likelihood over the set of parameters of M. Thus, BIC is commonly used to choose the best parametric model among the set of models with different number of parameters.

Following the same notations as in Sect. 8.5.2.4, under hypothesis H_0 we have:

$$\mathrm{BIC}_{H_0} = \log f_{X_i X_j}\left(\mathbf{X}_i \cup \mathbf{X}_j | \Theta_{f_{X_i X_j}}\right) - \lambda \frac{1}{2} \#(\Theta_{f_{X_i X_j}}) \log(N_i + N_j) \tag{8.32}$$

Likewise, under hypothesis H_1:

$$\begin{aligned} \mathrm{BIC}_{H_1} = &\log f_{X_i}\left(\mathbf{X}_i | \Theta_{f_{X_i}}\right) + \log f_{X_j}\left(\mathbf{X}_j | \Theta_{f_{X_j}}\right) \\ &- \lambda \frac{1}{2}\left(\#(\Theta_{f_{X_i}}) + \#(\Theta_{f_{X_j}})\right) \log(N_i + N_j) \end{aligned} \tag{8.33}$$

The BIC distance metric is then defined as:

$$d_{\mathrm{BIC}} = \mathrm{BIC}_{H_1} - \mathrm{BIC}_{H_0} \tag{8.34}$$

The above expression can be re-written in terms of d_{GLR} as:

$$d_{\mathrm{BIC}} = d_{\mathrm{GLR}} - \lambda \frac{1}{2}\left(\#(\Theta_{f_{X_i}}) + \#(\Theta_{f_{X_j}}) - \#(\Theta_{f_{X_i X_j}})\right) \log(N_i + N_j) \tag{8.35}$$

$$= d_{\mathrm{GLR}} - \lambda \frac{1}{2} \Delta \log(N_i + N_j) \tag{8.36}$$

where Δ is the difference between the number of free parameters of models in hypothesis H_1 and hypothesis H_0. From (8.36), BIC distance can be considered as a penalized GLR distance, with the penalty depending on the free parameter λ, number of parameters as well as number of observations. The selection of free parameter λ has been subject of constant study.

BIC is introduced for the case of speech and specifically for acoustic change detection and clustering by Chen and Gopalakrishnan [29], where the problem is formulated as that of model selection. In this paper, the authors introduced a tunable parameter λ in the penalty term which is used to improve the performance of the system for a particular condition in practice. This parameter therefore implicitly defines a threshold which needs to be tuned to the data and its correct setting has been subject of constant study [34,82,94,131,139]. Ajmera [8,9] proposes a method to cancel the penalty term by adjusting the number of free parameters in the models accordingly. The authors use a GMM with diagonal covariance matrices for each of the pdfs f_{X_i}, f_{X_j} and $f_{X_i X_j}$. By ensuring that the number of mixtures in $f_{X_i X_j}$ equals to the number of mixtures in f_{X_i} plus the number of mixtures in f_{X_j}, as a result $\Delta = 0$, and the penalty term is eliminated. In this case:

$$d_{\mathrm{BIC}} = d_{\mathrm{GLR}} \tag{8.37}$$

In Chen and Gopalakrishnan [29], it is shown that BIC value increases according to data size. This presents in general a problem when there is a big mismatch between clusters or windows with different data sizes. Thus, Perez-Freire [103] introduces the penalty weight which depends on the data size in order to achieve better robustness. Vandecatseyes [139] normalizes the BIC score by the total number of frames and shows that it consistently outperforms non-normalized BIC.

8.5.2.6 Cross Likelihood Ratio

The *Cross Likelihood Ratio* (CLR) measure was first used in Reynolds et al. [107] to compute the distance between two adapted speaker models and it was defined as:

$$d_{\text{CLR}} = \log\left(\frac{\mathscr{L}\left(\mathbf{X}_i \,|\, f_{X_i}\right)}{\mathscr{L}\left(\mathbf{X}_i \,|\, f_{\text{UBM}}\right)}\right) + \log\left(\frac{\mathscr{L}\left(\mathbf{X}_j \,|\, f_{X_j}\right)}{\mathscr{L}\left(\mathbf{X}_j \,|\, f_{\text{UBM}}\right)}\right) \tag{8.38}$$

where f_{UBM} is pdf of the *Universal Background Model* (UBM); f_{X_i}, f_{X_j} are pdfs of the adapted speaker models for speaker i and speaker j, respectively. The UBM is trained with a huge amount of audio data according to the gender (male, female) and the channel conditions. The speaker models are derived by adapting the UBM parameters with speaker speech data. The adaptation method often used is the Maximum A Posteriori (MAP) [50] adaptation. The CLR is commonly employed as distance metric in agglomerative speaker clustering systems including Barras et al. [19], Reynolds and Torres-Carrasquillo [110], and Sinha et al. [123].

8.5.2.7 Normalized Cross Likelihood Ratio

Normalized Cross Likelihood Ratio (NCLR) was presented as a distance measure between two speaker models [77]. Given two speaker models f_{X_i} and f_{X_j}, the NCLR distance is defined as:

$$d_{\text{NCLR}} = \frac{1}{N_i}\log\left(\frac{\mathscr{L}(\mathbf{X}_i \,|\, f_{X_i})}{\mathscr{L}(\mathbf{X}_i \,|\, f_{X_j})}\right) + \frac{1}{N_j}\log\left(\frac{\mathscr{L}(\mathbf{X}_j \,|\, f_{X_j})}{\mathscr{L}(\mathbf{X}_j \,|\, f_{X_i})}\right) \tag{8.39}$$

8.5.2.8 Other Distance Measures

8.5.2.8.1 Gish-Distance

Gish et al. [53] proposed a distance measure, which is referred to as Gish-distance in the literature. This distance is based on *likelihood ratio* with the assumption of multivariate Gaussian models and was used as clustering metric in [53] and [65]. van Leeuwen [78] employed Gish distance for agglomerative clustering in the TNO

speaker diarization system. Jin et al. [61] performed agglomerative clustering with Gish-distance and scaling heuristic to favor merging of consecutive segments.

8.5.2.8.2 Vector Quantization Distortion

In Mori and Nakagawa [94], the experimental results demonstrated superior performance of *vector quantization* (VQ) metric in both speaker segmentation and speaker clustering comparing to GLR and BIC. However, the database is too small (only 175 utterances) and restrictive (only clean speech) to deduce any conclusions.

8.5.2.8.3 XBIC

In Anguera [14], a XBIC metric, which is based on *cross-likelihood* between each data segment and the model trained on data from the other segment, is introduced for speaker segmentation and is shown to behave similar or better to BIC with reduction in computation.

8.5.2.8.4 Probabilistic Pattern Matching

Malegaonkar et al. [86] employed a probabilistic pattern matching approach for detecting speaker changes and studied different likelihood normalization techniques to improve the robustness of the proposed metric, and as a result better speaker segmentation was achieved comparing to BIC.

8.5.3 Speaker Segmentation

Speaker segmentation, with the aim to split the audio stream into speaker homogenous segments, is a fundamental process to any speaker diarization systems. While many state-of-the-art systems tackle the problem of segmentation and clustering iteratively, traditional systems usually perform speaker segmentation or *acoustic change point detection* independently and prior to the clustering stage. Various segmentation algorithms have been investigated in previous works, which can be categorized in one of the following groups: (1) *silence detection based methods*, (2) *metric-based segmentation*, and (3) *hybrid segmentation*.

8.5.3.1 Silence Detection Based Methods

Some of the speaker segmentation techniques are based on silence detection in speech signal. In these methods, it is supposed that there exists a silence region at

change points between speaker turns. The silence was detected either by a decoder [81] or directly by measuring and thresholding the audio energy [65, 98]. The segments are then generated by cutting the input at silence locations. However, the accuracy of such naive techniques is poor [65]. Moreover, the correlation between the existence of a silence in a recording and a change of speaker is arbitrary at most. Therefore, such techniques are usually used to detect hypothetical change points, which are then confirmed by more advanced techniques in the later stage.

8.5.3.2 Metric-Based Segmentation

The favored approach to speaker segmentation is to observe adjacent windows of data and calculating a distance metric between the two, then deciding whether the windows originated from the same or different speakers. The decisions generally base on a threshold/penalty term and this threshold is set empirically by using an additional development data. Various *metric-based segmentation* algorithms have been proposed in the literature, the differences among them lie mainly in the choice of distance metrics, the size of two windows, the time increment of the shifting of the two windows, and the threshold decisions.

8.5.3.2.1 Fixed-Size Sliding Window

In the pioneered work by Siegler et al. [121], the authors represented each window as a Gaussian and calculated the distance between the two distributions using the symmetric KL2 distance. To accomplish this, means and variances were estimated for a 2 s window placed at every point in the audio stream. When the KL2 distance between bordering windows reaches a local maximum, a new segment boundary is generated. The same framework was applied in Bonastre et al. [22] with the GLR as the distance metric and a tuned threshold to avoid missed detection to the detriment of false alarms. In Adami et al. [5], an initial speaker model was estimated from the small segment at the beginning of the conversation and the segment that has the largest GLR distance from the initial segment was used to train second speaker model. The segment boundaries are defined at the points where the GLR distances with respect to both speakers are equal; each segment in the conversation is assigned to the speaker with the smallest distance. Kim et al. [67] used DSD for speaker change detection with the covariances estimated for two sliding windows of 3 and 2.5 s overlapping. They showed that the DSD metric is more accurate than the BIC approach in presence of short segments, while both approaches are equivalent on long segments. In a more recent work, inspired by speaker verification techniques, a probabilistic pattern matching method with several likelihood normalization methods were investigated for speaker segmentation task in [86]. The proposed bi-lateral scoring scheme was shown to be more effective than BIC and XBIC, mainly due to the inclusion of score normalization techniques.

8.5.3.2.2 Variable-Size Sliding Window

Later, Chen and Gopalakrishman [29] formulated the problem of speaker change detection as a model selection problem and applied BIC for this purpose. This technique looks for potential change points in a window of frames by testing two hypotheses: the first hypothesis assumes the data in the window belong to one speaker and therefore is better represented by that speaker distribution, on the other hand the second hypothesis assumes that there are two different speakers, hence the data are better modelled by two different distributions. In case there is no change point detected within the window, its size is increased by a certain number of frames depending on the algorithm and the process is repeated. Tritschler and Gopinath [131] suggested another variable window scheme in which the size of the window is increased adaptively in contrast to a fixed amount as in Chen and Gopalakrishnan [29]. They also devised some rules to eliminate some of the BIC tests in the window, when they correspond to locations where the detection of a boundary is very unlikely. These heuristics make the algorithm faster and give importance to detecting short changes. In Sivakumaran et al. [124] and Cettolo and Vescovi [28], by significantly reducing the number of operations involved in the estimation of the means and covariance matrices, the segmentation process were sped up. In Roch and Cheng [111], a MAP-adapted version of the models was presented, which allows for shorter change points to be found at the cost of being slightly worse than EM-trained models when longer hypothesis windows are used. A notable variation to BIC has been proposed by Ajmera and Wooters in [8], in which the authors fixed the number of parameters between the two BIC hypotheses so as to eliminate the need for tuning the BIC penalty term.

8.5.3.2.3 Multi-Step Segmentation

There are also works which attempt to make the detection procedure faster by applying a distance measure prior to BIC. DIST-BIC [33, 34] is a work in this direction. A log-likelihood ratio (LLR) based distance computation prior to BIC was proposed in this work, which is faster than BIC. Then, only selected change points are passed through the BIC test. In Zochova et al. [163], the same framework was used with some modifications in speaker change candidate detection and speaker change position location. They reported better results in a majority of tests. Also in this direction, Zhou and Hansen [160] proposed applying T^2-statistics prior to BIC. The authors claimed to improve the algorithm speed by an order of 100 compared to Chen and Gopalakrishnan [29] without sacrifice the overall performance. Lu and Zhang [83] applied KL2 distance on *line spectrum pair* (LSP) frequency features; the speaker change detection scheme was able to meet the requirement of real-time processing in multimedia applications. Vandecatseye [139] used a measure called *normalized log likelihood ratio* (NLLR) to generate potential change points in the first stage and then used normalized BIC in second stage to eliminate false alarm turns. All of these algorithms perform in a bottom-up manner where there are many

short speaker turns in the first step which will be eliminated subsequently in the second step. However, Wang [149] proposed a method which perform in top-down manner. A long sliding window was first used to segment a long audio stream into shorter sub-segments, and sequential divide and conquer segmentation was applied to each sub-segment with shorter window to detect the remaining change points. Both stages used BIC as the distance metric. In Gangadharaiah et al. [49], a two-speaker segmentation algorithm was performed in two steps. In the first step, a standard approach for segmentation was applied using GLR with same size adjacent windows, fixed step shifting. In the second step, several segments were selected to train a GMM for each speaker and the rest were assigned to either speaker with a *maximum likelihood* (ML) approach.

8.5.3.3 Hybrid Segmentation

Liu and Kubala [81] introduced a two-stage hybrid segmentation system combining model-based and metric-based approach. The output of a phone-based decoder was used as the initial segments and a new penalized GLR criterion was employed to accept/reject change-points previously found. Kemp at al. [65] chopped the input signal into short segments of 1 s, then performed bottom-up clustering using Gish distance until a predetermined number of clusters remained. GMMs were trained for each cluster and a model-based segmenter was then applied.

8.5.3.4 Segmentation Evaluation

In evaluating segmentation performance, two kinds of error measures are commonly computed, *false alarm rate* (FAR) and *miss detection rate* (MDR):

$$\text{FAR} = \frac{\text{Number of false alarms}}{\text{Number of detected change points}} \tag{8.40}$$

$$\text{MDR} = \frac{\text{Number of miss detections}}{\text{Number of actual change points}} \tag{8.41}$$

where a *false alarm* refers to a change point is detected but it does not exist, a *miss detection* refers to an existing change point but is not detected by the algorithm. On the other hand, one may use the *recall* (RCL) and *precision* (PRC) defined as:

$$\text{RCL} = 1 - \text{FAR} \tag{8.42}$$

$$\text{PRC} = 1 - \text{MDR} \tag{8.43}$$

In order to consider the trade-off between these two metrics, the F measure can be used:

$$F = \frac{2 \times \text{PRC} \times \text{RCL}}{\text{PRC} + \text{RCL}} \tag{8.44}$$

8.5.4 Speaker Clustering

8.5.4.1 Agglomerative Hierarchical Clustering

Most state-of-the-art speaker diarization systems employ *agglomerative hierarchical clustering* (AHC) architecture, also known as *bottom-up clustering*, where the systems start with an overdetermined number of segments/clusters and via merging procedures to converge to the optimum number of clusters determined by some stopping criteria.

8.5.4.1.1 Step-By-Step Speaker Segmentation and Clustering

This is the classical approach where change point detection is typically used to segment the recording into speaker homogeneous segments and then these segments are grouped together according to a distance measure until a stopping criterion is satisfied.

The work of Jin et al. [61] was probably one of the earliest research done in speaker clustering with intention for speaker adaptation in *automatic speech recognition* (ASR) systems. After segmentation, the system built a distance matrix using *Gish-distance* based on the Gaussian models of the acoustic segments and hierarchical clustering was performed on this distance matrix, in which the distance between consecutive segments was scaled by a factor to increase the probability of merging these segments. As stopping criterion, the optimal clustering was selected by minimizing the within-cluster dispersion with some penalty against too many clusters. However, no systematic way to deduce the optimal value of the penalty term was proposed in this work. With the same purpose of speaker adaptation, in Siegler et al. [121], the KL2 distance was used as a distance metric. The authors showed that the KL2 distance works better than the Mahalanobis distance for speaker clustering. In this work, the stopping criterion was determined with a merging threshold, which was presumably tuned on development data set however no training procedure was specified. Solomonoff et al. [125] used the GLR distance matrix for speaker clustering. The authors proposed a cluster purity metric to evaluate the quality of a partition, which can also be used to determine the appropriate number of clusters. However, this metric requires knowing the true speaker of each segments, thus a method to estimate cluster purity without true labels was also presented in the paper. The estimation method involved a tunable parameter, but the authors did not clearly indicate how to obtain this parameter value. In Tsai et al. [134], they also used the same metric to determine number of speakers, however with an entirely different inter-cluster distance measurement. Instead of training models for each speaker clusters as classical measurement, they projected the segments/clusters into a speaker reference space, in which the distances between segments/clusters are claimed to be more effective and reliable. Although fairly good performance has been obtained, they also raised concern

about the correlation between speaker reference bases, which ideally should be statistically independent with each other. Jin et al. [62] also used GLR metric in their work with the modification in speaker models. Instead of training each speaker model independently, they constructed a UBM from all the speech segments of that recording then used MAP adaptation for each individual speaker.

Chen and Gopalakrishnan [29] introduced BIC metric for speaker segmentation and clustering. In this work, starting from each individual segment as a cluster, hierarchical clustering was performed by calculating the BIC measure for every pair of clusters and merging the two clusters with the highest BIC measure. The clustering is stopped when no two clusters resulted into an increase in BIC measure, when merged. Zhou and Hansen [160] proposed a multi-step segmentation approach using T^2-statistic to select potential change points and validate with BIC to speed up the segmentation procedure. Then a GMM classifier was employed to automatically label the segments as male or female speech, and bottom-up clustering was performed on the segments of each gender independently, with BIC as distance metric and stopping criterion. It was shown that with gender labeling, the resulting clusters have higher purity in comparison with no gender labeling. This algorithm has been applied for audio indexing tasks in [55]. In Cassidy [27], Mahalanobis was used as cluster distance metric to merge all segments longer than 1.5 s and BIC was used as stopping criterion. Once speaker clustering has been performed, a Gaussian model was then trained on each cluster and these models were used to classify all speech segments. The similar framework was employed in van Leeuwen [78] with BIC for speaker segmentation, stopping criterion and Gish distance for agglomerative clustering. Zhu et al. [161] applied Viterbi re-segmentation after the initial segmentation with *Gaussian divergence measure*. Then, a two-stage clustering method was performed, with BIC agglomerative clustering preceding a speaker identification module. The boundaries of speech segments are kept unaltered during the clustering process.

In some later works, the additional re-segmentation step was implemented after the clustering procedure, to refine the segment boundaries. In [19], the authors used BIC as both the distance metric and stopping criterion with the addition module of speaker identification (SID) clustering. The system first classified each cluster for gender and bandwidth and used MAP adaptation to derive speaker models from each cluster. In SID clustering process, agglomerative clustering was performed separately for each gender and band condition, the speaker models were compared using a metric between clusters named cross likelihood distance [107]. It was shown that with the addition of SID clustering, the diarization error rate could be reduced nearly 50 % relatively. In Kim et al. [67], a multi-stage approach was proposed which includes: speaker segmentation using DSD metric, initial clustering with BIC as distance metric, clustering by HMM models likelihood scores, and finally HMM-based re-segmentation. The combined method was shown to outperform any individual approach. The LIUM speaker diarization system [60] was also based upon a standard framework composed of three modules: signal split into small homogeneous segments with GLR metric, speaker clustering without changing the boundaries with BIC metric, and boundaries adjustment with Viterbi decoding.

Other than HMM and Viterbi, Reynolds and Torres-Carrasquillo [110] employed the iterative re-segmentation using GMM for frame-based classification with a smoothed window of 100 frames. Moreover, they also included non-speech models in the re-segmentation step.

8.5.4.1.2 Iterative Speaker Segmentation and Clustering

Typically, in this framework, initial clusters are obtained by some initialization procedures. Models are then trained on these clusters and Viterbi decoding is performed to identify when the speaker changes occur as well as the hypothesized speaker identity of each segment. The procedure is repeated: the latest segmentation is used to train the hypothesized speaker models, then Viterbi decoding is run to perform speaker segmentation and speaker clustering. Figure 8.4 shows the typical structure of the state-of-the-art single channel speaker diarization systems using iterative agglomerative hierarchical clustering approach.

In Gauvain et al. [51], the authors proposed an iterative GMM clustering method which uses an objective function based on penalized log-likelihood. For each iteration, all possible pairs of segments are considered for merging and its corresponding likelihood loss is calculated. Eventually, the pair with the smallest loss is merged and the GMM statistics were reevaluated. The process is reiterated until the likelihood loss crosses a specified threshold, then the data is segmented using a Viterbi decoder with the newly estimated GMMs. The whole process is repeated until the segmentation converges or a maximum number of iterations are reached. Two parameters are introduced in this function to penalize number of segments and number of clusters. The function therefore could be used both to determine which clusters should be merged and to determine when to stop merging. However, the selection of parameters is too ad-hoc and there are two parameters to tune, this method is coupled with robustness issue. Sinha et al. [123] also used a similar *iterative agglomerative hierarchical clustering* (IAHC) scheme with the addition of *speaker identification* (SID) clustering after IAC. In the SID clustering phase, CLR metric is employed to select the closest pair of clusters to be merged and a threshold is defined to stop the merging process.

The paper by Ajmera and Wooters [9] was probably the first to suggest an iterative, agglomerative clustering technique based on a HMM framework. A uniformly initial segmentation is used to train speaker models that iteratively decode and retrain on the acoustic data. Pairs of closest clusters are merged in successive iterations and merging stops automatically using a threshold-free BIC metric. Later, in the ICSI-SRI fall 2004 diarization system, Wooters et al. [152] used the same framework with the introduction of Viterbi segmentation likelihood scores as stopping criterion. The Viterbi stopping criterion was reported to be slightly better than BIC which is contributed by the reduction in speaker error rate. In Anguera et al. [18], they improved this framework with an addition of purification algorithm to split clusters that are not acoustically homogeneous.

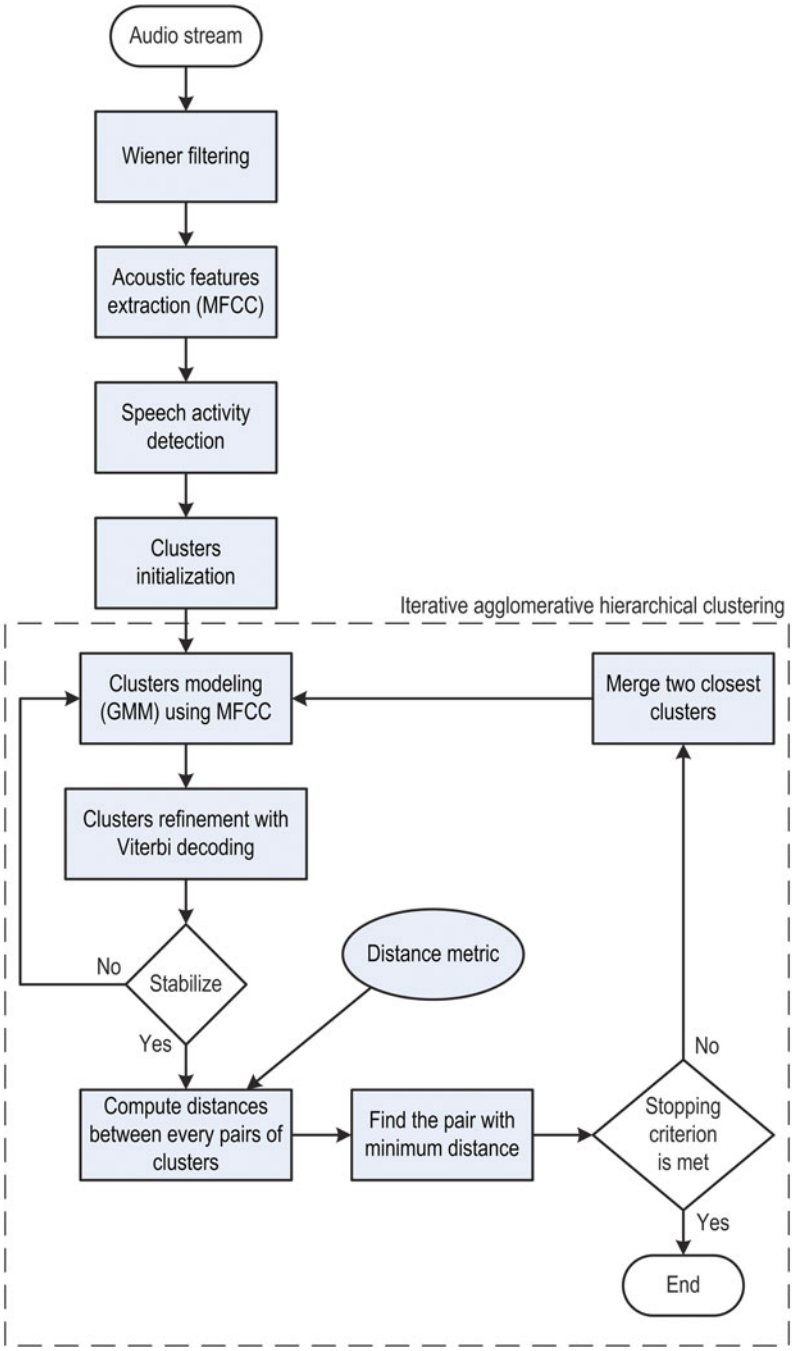

Fig. 8.4 Typical structure of the state-of-the-art single channel speaker diarization systems

8.5.4.1.3 Information Bottleneck

Agglomerative Information Bottleneck (aIB) is another bottom-up algorithm used to perform speaker diarization. The goal of the aIB system is to iteratively merge uniform short segments $S = s_1, s_2, \ldots, s_N$ into clusters $C = C_1, C_2, \ldots, C_K$ which simultaneously maximize the mutual information $I(Y, C)$ of a set of relevance variables Y and a set of clusters C, while minimizing the mutual information $I(C, X)$ of C and a set of segments S, as shown in Eq. (8.45). The merging continues until the stopping criterion is met. After which, Viterbi decoding is performed in order to determine the segment boundaries.

$$max \left[I(Y, C) - \frac{1}{\beta} I(C, X) \right] \tag{8.45}$$

where Y is a set of components of a background GMM trained on the entire audio recording, and β is a Lagrange multiplier. Thus, Eq. (8.45) is used to determine a cluster representation C which is useful for describing the relevance variables Y (maximize $I(Y, C)$) and simple (minimize $I(C, X)$). The aIB is more computationally efficient than the HMM-GMM speaker diarization system since new models are not trained for each potential merging of two clusters. Instead, for the aIB framework subsequent statistics are taken to be averages of previously defined statistics. Speaker diarization systems which employed aIB are predominantly implemented by Vijayasenan et al. [140–145].

8.5.4.1.4 Multi-Stream Clustering

With the available of multi-channel recordings, the time delays between microphone pairs can be computed. In [39], Ellis and Liu employed spectral subspace approach to cluster the delay feature vectors into different groups where each group represents individual speaker. However, the system missed many speaker turns which incurred miss detection errors and resulted in high overall error rate. In Pardo et al. [99], the same clustering framework as in Ajmera and Wooters [9] was used with time delay feature in place of acoustic feature. The system was compared to that of [39] and significant improvement was reported. However if comparing these results with the results obtained from the same systems using standard acoustic feature, there is still a big gap to cover. Luque et al. [84] analyzed the TDOA distribution of a recording and exploited the most likely and stable pairs of TDOA to obtain an initial clustering of speakers. An iterative agglomerative clustering algorithm similar to [9] was then performed with MFCC as features. The authors reported better performance comparing to uniform initialization of clusters. In Anguera [16], the inter-delay features and acoustic features were cleverly combined in a multi-stream HMM framework and the performance is greatly improved. In this framework, each feature stream is assigned a weight which reflects the relative contribution of individual feature stream; the weights were learned from development data. Later in [17],

the same authors proposed an automatic weighting for the combination of these two feature streams. The scheme was later used in ICSI RT07s Speaker Diarization System [153] and this is the state-of-the-art system thus far. Following the same clustering framework, the AMIDA speaker diarization system [79], in contrast to uniform initialization, started with typically 40 initial clusters by performing segmentation and clustering using BIC. The system employed CLR as clustering distance measure using both cepstral and delay features. The weight of each feature stream was fixed, with higher contribution given to acoustic features.

Apart from time delay features, other features may also be combined with conventional spectral features in multi-stream speaker diarization system. Vinyals and Friedland [147] proposed the use of *modulation spectrogram* [68] as an additional stream of features to the commonly used MFCCs. In this work, the clustering framework follows the ICSI agglomerative clustering approach [9, 18, 153] with fixed weighting for each feature stream. In [46, 48], the authors investigated a large set of 70 prosodic and long-term features and applied Fisher criterion to rank these features by their ability to discriminate speakers. It is shown in the paper that the combination of MFCC features with the additional top-ten ranked prosodic and long-term features leads to improvement in terms of *diarization error rate*.

8.5.4.2 Divisive Hierarchical Clustering

Divisive hierarchical clustering, also known as top-down clustering, starts with very few clusters and proceed to split the clusters iteratively until the desired number of clusters is reached. In the current literature there are few systems following this clustering framework.

A top-down split-and-merge speaker clustering frame work was proposed in Johnson and Woodland [64] to enhance the accuracy of ASR in broadcast news by improving the unsupervised speaker adaption. The clustering algorithm starts with one node consisting of the whole speech recording. At each stage, a node is considered to be split into four child nodes if some segments belong to that node might move to other nodes using the *maximum likelihood* (ML) criterion. The splitting is continued until the algorithm converges or the maximum number of iterations is reached. At each stage of splitting, clusters that are very similar to each other are allowed to merge. Two different implementations of the algorithm were proposed: one was based on direct maximization of MLLR and one was based on AHS metric. In Johnson [63], the similar framework with AHS distance metric was applied for speaker diarization with the same stopping criterion as proposed in Solomonoff et al. [125].

In Meigner et al. [89], an iterative approach combining both segmentation and clustering in a top-down manner named evolutive HMM (e-HMM) was proposed. Initially, the system starts with one HMM trained on all the acoustic data available. The best subset features of this model (in terms of maximum likelihood scores) are taken out to train a new model using MAP adaptation. According to the subset selected, a segmentation is performed using Viterbi decoding. This process

is repeated until the gain in likelihood score is insignificant, which is controlled with a tunable parameter. This parameter significantly influences the segmentation error as reported in the paper. In Anguera and Hernando [13] a similar approach was followed and a repository model was further introduced, which showed an improvement of 20 % relatively.

8.5.4.3 Other Approaches

8.5.4.3.1 Self Organizing Map

In [72], Lapidot presented an approach for speaker clustering based on *Self Organizing Map* (SOM) given a known number of speakers. In this approach, SOM is used as likelihood estimators for speaker model and BIC is applied for estimation of the number of clusters.

8.5.4.3.2 Genetic Algorithm

In Tsai and Wang [132], they formulated the problem of speaker clustering as that of maximizing the overall within-cluster homogeneity. The within-cluster homogeneity is defined as the likelihood probability that a cluster model, trained using all the utterances within a cluster, matches each of the within-cluster utterances. This probability is maximize using genetic algorithm with initial random cluster assignment and iterative evaluation of the likelihood and mutation. In order to select the optimum amount of clusters they used BIC computed on the resulting models.

8.5.4.3.3 Variational Bayesian

In [136, 137], Valente and Wellekens explored the use of Variational Bayesian (VB) learning, which has the capacity of model parameter learning and model complexity selection at the same time, for speaker clustering. With the proposed VB approach, the initial speaker models could be modeled as GMMs with any number of Gaussians, VB automatically prunes together with the cluster number, the best Gaussian model at the same time, resulting in smaller models where few observations are available and in bigger models where more observations are available.

8.5.4.3.4 Dirichlet Process Mixture Model

In the previous works, the model learning methods (EM, ML, MAP) require the model space (such as number of mixtures, components, states etc.) is known a priori. Recently, Valente [135] proposed the use of infinite models for speaker clustering,

in which the segmentation is obtained through a Dirichlet Process Mixture Model (DPMM). DPMM is a flexible model of unknown complexity with a prior on the parameters follow Dirichlet Process [40] which avoids fixing the number of observation modes. The experiments on broadcast news data showed improvements over ML/BIC, MAP/BIC and VB. In [44], Fox et al. extended the original work on hierarchical Dirichlet process hidden Markov model (HDP-HMM) [129] and apply this framework for speaker diarization on the NIST meeting database. The reported result was comparable to that of the state-of-the-art system [153], which use agglomerative BIC clustering, in NIST Rich Transcription Evaluations 2007.

8.5.4.4 Multiple Systems Combination

The speaker diarization systems presented thus far use either top-down or bottom-up technique for clustering. There are some works on algorithms to combine multiple systems and obtain an improved speaker diarization. In Tranter [130], the author presented a cluster-voting scheme designed to reduce the diarization error rate by combining information from two different diarization systems. Improvements were shown on broadcast news database when combining two bottom-up systems and two top-down systems. In Moraru et al. [92, 93], two strategies for systems combination were presented, those are: hybridization strategy, and merging strategy. The hybridization strategy consists of using segmentation results of the bottom-up system to initialize the top-down system. This solution associates the advantages of longer and quite pure segments of the agglomerative hierarchical approach with the HMM modeling and decoding power of the integrated approach. The merging strategy proposes a matching of common resulting segments followed by a re-segmentation of the data to assign the non-common segments.

8.5.5 Online Speaker Clustering

Online speaker diarization systems, though sharing many similarities with their offline counterparts, have some distinctive components which are considered to be more important for online learning, these include: (1) *novelty detection* where new speakers are detected, and (2) *incremental learning* where speaker models are updated adaptively with new observations. Given the major considerations for real-time systems is the latency, some techniques discussed in this section are the scaled-down versions of the offline techniques, while others are specifically devised for real-time processing. The majority of online speaker diarization systems adopted the conventional leader-follower clustering method [36], its structure is depicted in Fig. 8.5.

Segmentation. Standard model-based speech activity detection approach was taken in Markov and Nakamura [87, 88] to obtain speech segments, where the frame

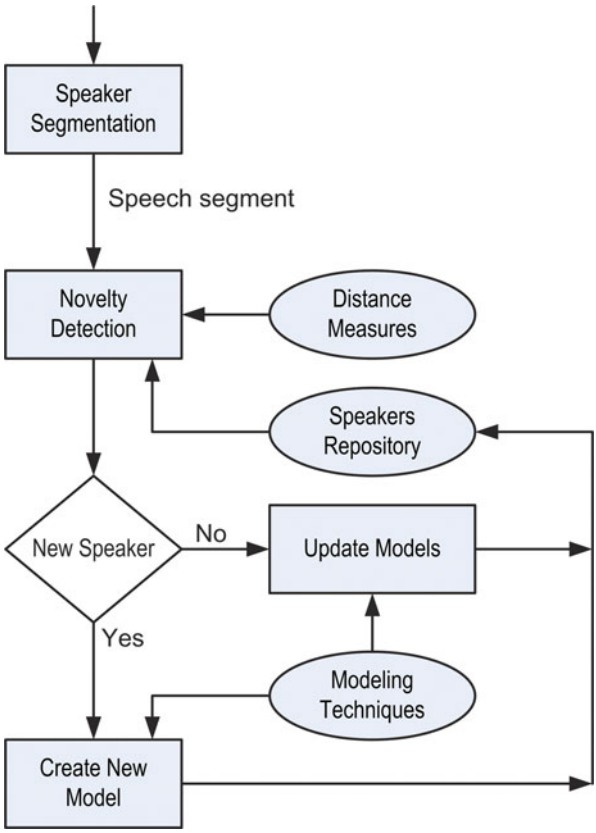

Fig. 8.5 Sequential clustering architecture with leader-follower clustering method

classification labels were smoothed with two median filters and duration constraints were applied to decide the start and end points of these segments. However, a key parameter named *decision time* (DT), which is essentially the latency time, was responsible to make decisions, e.g. the system outputs the decisions at DT time after the speech data are available, regardless the length of the speech segment. On the other hand, a simple energy-based SAD with duration constraint was employed in [52] to label initial speech segments. These segments were then confirmed or rejected by a model-based detector with gender identification and the system made decisions every times when the segment end points are detected.

Novelty Detection. This component is responsible for detecting unseen speakers which are not in the current repository and is considered to be essential for online speaker diarization system. To this end, the authors of [52] applied an open-set speaker identification technique where each short segment is verified against all existing speaker models using likelihood ratio scoring. If none of the models is matched, a new speaker model is created by adapting the corresponding gender dependent UBM to this speaker data, otherwise the existing model is updated with

the newly available observations. In [80], the GLR metric was used to measure the distance between the new speech segment and all models in the repository. The pair with minimum distance was then selected and compared to a pre-defined threshold to confirm whether the segment belongs to this speaker. If it does not, a Gaussian model is estimated from the speech segment and new speaker model is added to the current database. Also based on likelihood ratio, Markov and Nakamura [87] performed a hypothesis test with two hypotheses: H_0 if the new segment belongs to an old speaker, and H_1 if it belongs to an unseen speaker. Three different approaches to compute the likelihood corresponding to hypothesis H_0 was presented in the paper. In their later work [88], inspired by speaker verification research, score normalization techniques were suggested to improve the robustness of novelty detection in terms of speaker genders and number of speakers in the database. In a rather different approach, Koshinaka et al. [70] employed BIC for model selection together with an ergodic HMM for this purpose.

Speaker Modeling. Various speaker modeling techniques have been discussed in Sect. 8.5.1, where GMM was widely adopted in many systems. However, traditional EM algorithm to estimate the parameters of GMM is relatively computational expensive for online system. To learn the speaker model rapidly and from limited available data, Geiger et al. [52] trained gender dependent UBMs and take advantage of MAP adaptation to quickly adapt the corresponding UBM with the speaker data. Alternatively, incremental versions of EM were proposed in [95] and some online variants followed later [115, 157], the techniques were then applied in Markov and Nakamura [87, 88] to estimate the speaker models. On the other hand, Koshinaka et al. [70] modeled each speaker as a state of an ergodic HMM, each state is a GMM. The model parameters are updated online with variational Bayesian learning algorithm.

8.5.5.1 Speaker Clustering Evaluation

Consider N_s speakers that are clustered into N_c groups, where n_{ij} is the number of frames in cluster i spoken by speaker j, n_{c_i} is the number of frames in cluster i, n_{s_j} is the number of frames spoken by speaker j, and N is the total number of frames.

8.5.5.1.1 Average Cluster Purity

The *average cluster purity* (acp) [7] gives a measure of how well a cluster is limited to only one speaker; it reduces when a cluster includes segments from two or more speakers. The acp is based on cluster purity which is defined as:

$$p_{c_i} = \sum_{j=1}^{N_s} \frac{n_{ij}^2}{n_{c_i}^2} \tag{8.46}$$

where p_{c_i} is the purity of cluster i. Then the acp is computed as:

$$\text{acp} = \frac{1}{N} \sum_{i=1}^{N_c} p_{c_i} n_{c_i} \tag{8.47}$$

8.5.5.1.2 Average Speaker Purity

On the other hand, the *average speaker purity* (asp) [7] gives a measure of how well a speaker is limited to only one cluster; it reduces when speech of a single speaker is split to more than one cluster. The asp is based on the speaker purity:

$$p_{s_j} = \sum_{i=1}^{N_c} \frac{n_{ij}^2}{n_{s_j}^2} \tag{8.48}$$

where p_{s_j} is the purity of speaker j. The asp is computed as:

$$\text{asp} = \frac{1}{N} \sum_{j=1}^{N_s} p_{s_j} n_{s_j} \tag{8.49}$$

8.5.5.1.3 K Measure

To balance the trade off between acp and asp, as well as to facilitate comparison between systems, Ajmera [7] propose the K measure, which is a geometrical mean of acp and asp:

$$K = \sqrt{\text{acp} * \text{asp}s} \tag{8.50}$$

8.5.5.1.4 Rand Index

The *Rand index* [106] is a widely used measure for comparing partitions. It gives the probability that two randomly selected frames are from the same speaker but grouped in different clusters, or the two frames are in the same cluster but from different speakers. Rand index is defined as:

$$R = \frac{1}{\binom{N}{2}} \left[\frac{1}{2} \left(\sum_{i=1}^{N_c} n_{c_i}^2 + \sum_{j=1}^{N_s} n_{s_j}^2 \right) - \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} n_{ij}^2 \right] \tag{8.51}$$

Rand index value changes from 0 to 1. The lower the index, the higher the agreement is between two partitions. However, it does not provide any information on how the partitions are distributed and how the two partitions are related.

8.6 Speaker Diarization Evaluation

The standard performance metric of the speaker indexing and diarization systems is the *diarization error rate* (DER). To evaluate the performance, an optimum mapping from the reference speakers in the conversation to the system speakers of the system should be found. The criterion for this mapping optimality is the percentage of the speech parts which are common to both the reference speaker and the system speaker. This optimality metric is calculated for all segments and all speakers. The mapping should map each reference speaker to at most one system speaker and vice versa. Once the optimal mapping is found, the DER is then evaluated as a time-based score which calculates the percentage of speaker time which is not mapped correctly to a reference speaker.

$$\mathrm{DER} = \frac{\sum_s \mathrm{dur}(s).\left(\max\left(N_{\mathrm{ref}}(s), N_{\mathrm{sys}}(s)\right) - N_{\mathrm{correct}}(s)\right)}{\sum_s \mathrm{dur}(s).N_{\mathrm{ref}}(s)} \tag{8.52}$$

where s is the longest continuous segments for which the reference and system speakers do not change, $\mathrm{dur}(s)$ is the duration of s, $N_{\mathrm{ref}}(s)$ is the number of reference speakers in s, $N_{\mathrm{system}}(s)$ is the number of system speakers in s and $N_{\mathrm{correct}}(s)$ is the number of mapped reference speakers which match the system speakers.

8.7 Databases for Speaker Diarization in Meeting

In this section, we list some of the available databases for meeting recordings in which the speaker segments are accurately transcribed to serve the need for speaker diarization task:

- **The ISL Meeting Corpus** [1]: 104 meetings with a total of 103 h. Each meeting lasts an average of 60 min, with an average of 6.4 participants.
- **The ICSI Meeting Corpus** [2]: 75 meetings collected during the years 2000–2002. The recordings range in length from 17 to 103 min, but generally about 1 h each. There are a total of 53 unique speakers in the corpus. Meetings involve from 3 to 10 participants, averaging 6.
- **NIST Meeting Pilot Corpus** [3]: 19 meetings collected between 2001 and 2003. Approximately 15 h of data are recorded simultaneously from multiple microphones and video cameras.

- **The AMI Meeting Corpus** [4]: 100 h of meeting recordings. The meetings are recorded in English using three different rooms with different acoustic properties, and include mostly non-native speakers.

8.8 Related Projects in Meeting Room

The Interactive Multimodal Information Management (IM2) aims at the study of multimodal interaction, covering a wide range of activities and applications, including the recognition and interpretation of spoken, written and gestured languages, computer vision, and the automatic indexation and management of multimedia documents. One of the most important and challenging applications is Smart Meeting Management. The overall objective of this application is the construction of a demonstration system to enable structuring, browsing and querying of an archive of automatically analysed meetings, which are captured from rooms equipped with multimodal sensors including: close-talk microphones, distant microphones, microphone arrays as well as cameras.

The Computers In the Human Interaction Loop (CHIL) aims at improving the interactions between users and computers by making computers more usable and receptive to the user's needs and realizing computer services that are delivered to people in an implicit, indirect and unobtrusive way. Several intelligent meeting rooms with audio and video sensors are built where data is collected and research is performed on the lecture-type meetings.

The project Augmented Multi-party Interaction (AMI) focuses on enhancing the productivity of meetings by changes in technologies and changes in business processes. The AMI Consortium studies the human behaviour in meetings using advance signal processing, machine learning models and social interaction dynamics. Within the scope of the project, Consortium members have developed a very large database of pre-processed meeting recordings of multiple sources of information (contained in audio, video and images captured). The actions, words and all data (slides, white board drawings and hand written notes) associated with a set of scripted meetings are captured using highly instrumented meeting rooms.

8.9 NIST Rich Transcription Benchmarks

With the goal of creating recognition technologies that will produce transcriptions which are more readable by humans and more useful for machines, the National Institute for Standards and Technology (NIST) has been organizing the Rich Transcription evaluation series since 2002. In recent years, the attention has been shifted toward meetings environment.

Table 8.1 Summary of recent NIST RT evaluations

	Source type		Meeting type		Tasks			
	Audio	Video	Lect	Conf	SAD	SPKR	STT	SASTT
RT 2005	x		x	x	x	x	x	
RT 2006	x		x	x	x	x	x	
RT 2007	x			x		x	x	x
RT 2009	x	x		x		x	x	x

The datasets used in the meetings evaluations are contributed by various recording sites including CMU, ICSI, LDC, NIST, AMI, VT, EDI, IDI and TNO. Two types of meetings are recorded: lecture (Lect) and conference (Conf). There are several main tasks in the evaluations:

- Speech Activity Detection (SAD)
- Speaker Diarization (SPKR)
- Speech-to-text (STT)
- Speaker Attributed Speech-to-text (SASTT): which is essentially the combination of SPKR and STT tasks.

These tasks are further divided into several conditions such as: individual head microphone (IHM), single distant microphone (SDM), multiple distant microphones (MDM), all distant microphones (ADM) which including source localization arrays. We summarize the recent evaluations in terms of data type and core tasks in Table 8.1.

8.10 Summary

The chapter has given an introduction to speaker diarization system in general and diarization in meetings in particular. The presentation focuses on off-line speaker diarization systems with hierarchical clustering framework as these approaches are thus far the most popular in the literature. With recent advances in speaker recognition techniques, and particularly with the success of the total variability approach, it is expected to see a shift from the current state-of-the-art IAHC framework to a more suitable clustering framework incorporating these latest breakthroughs. All in all, there are still many remaining challenges and issues to be addressed, such as handling of overlap speech, improving real-time speaker diarization systems or multi-sessions speaker diarization (speaker attribution).

References

1. The ISL Meeting Corpus (2004), https://catalog.ldc.upenn.edu/LDC2004S05. Accessed 25 Aug 2014
2. The ICSI Meeting Corpus (2004), https://catalog.ldc.upenn.edu/LDC2004S02. Accessed 24 Aug 2014
3. NIST Meeting Room Pilot Corpus (2004), https://catalog.ldc.upenn.edu/LDC2004S09. Accessed 24 Aug 2014
4. The AMI corpus (2007), http://groups.inf.ed.ac.uk/ami/download/. Accessed 25 Aug 2014
5. A.G. Adam, S.S. Kajarekar, H. Hermansky, A new speaker change detection method for two-speaker segmentation, in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, vol. 4 (2002), pp. 3908–3911
6. A.G. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S.S. Kajarekar, N. Morgan, S. Sivadas, Qualcomm-ICSI-OGI features for ASR, in *Interspeech* (2002)
7. J. Ajmera, H. Bourlard, I. Lapidot, I. McCowan, Unknown-multiple speaker clustering using HMM, in *Interspeech* (2002)
8. J. Ajmera, I. McCowan, H. Bourlard, Robust speaker change detection. IEEE Signal Process. Lett. **11**(8), 649–651 (2004)
9. J. Ajmera, C. Wooters, A robust speaker clustering algorithm, in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding, 2003 (ASRU'03)* (2003), pp. 411–416
10. J. Allen, How do humans process and recognize speech? IEEE Trans. Speech Audio Process. **2**(4), 567–577 (1994)
11. X. Anguera, BeamformIt acoustic beamformer (2009), http://www.xavieranguera.com/beamformit/. Accessed 24 Aug 2014
12. X. Anguera, M. Aguilo, C. Wooters, C. Nadeu, J. Hernando, Hybrid speech/non-speech detector applied to speaker diarization of meetings, in *IEEE Odyssey 2006: The Speaker and Language Recognition Workshop* (2006), pp. 1–6
13. X. Anguera, J. Hernando, Evolutive speaker segmentation using a repository system, in *Proceedings of International Conference on Speech and Language Processing*, Jeju Island, 2004
14. X. Anguera, J. Hernando, Xbic: real-time cross probabilities measure for speaker segmentation. University of California Berkeley, ICSIBerkeley Technical Report (2005)
15. X. Anguera, C. Wooters, J. Hernando, Automatic cluster complexity and quantity selection: towards robust speaker diarization, in *Machine Learning for Multimodal Interaction* (Springer, Berlin, 2006), pp. 248–256
16. X. Anguera, C. Wooters, J. Pardo, Robust speaker diarization for meetings: ICSI RT06s evaluation system, in *Ninth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 2006)
17. X. Anguera, C. Wooters, J. Pardo, J. Hernando, Automatic weighting for the combination of TDOA and acoustic features in speaker diarization for meetings, in *Proceedings of ICASSP* (2007), pp. 241–244
18. X. Anguera, C. Wooters, B. Peskin, M. Aguiló, Robust speaker segmentation for meetings: the ICSI-SRI spring 2005 diarization system, in *Machine Learning for Multimodal Interaction* (Springer, Berlin, 2006), pp. 402–414
19. C. Barras, X. Zhu, S. Meignier, J.L. Gauvain, Improving speaker diarization, in *RT-04F Workshop* (2004)
20. M. Ben, M. Betser, F. Bimbot, G. Gravier, Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted GMMs, in *Eighth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 2004)
21. F. Bimbot, L. Mathan, Text-free speaker recognition using an arithmetic-harmonic sphericity measure, in *Third European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 1993)

22. J.F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, C. Wellekens, A speaker tracking system based on speaker turn detection for NIST evaluation, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000 (ICASSP'00)*, vol. 2 (2000), pp. 1177–1180
23. S. Bozonnet, N. Evans, C. Fredouille, The lia-eurecom RT'09 speaker diarization system: enhancements in speaker modelling and cluster purification, in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (2010), pp. 4958–4961. doi:10.1109/ICASSP.2010.5495088
24. J. Campbell et al., Speaker recognition: a tutorial. Proc. IEEE **85**(9), 1437–1462 (1997)
25. W. Campbell, D. Sturim, D. Reynolds, Support vector machines using GMM supervectors for speaker verification. IEEE Signal Process. Lett. **13**(5), 308–311 (2006). doi:10.1109/LSP.2006.870086
26. G.C. Carter, A.H. Nuttall, P.G. Cable, The smoothed coherence transform. Proc. IEEE **61**(10), 1497–1498 (1973)
27. S. Cassidy, The Macquarie speaker diarization system for RT04s, in *NIST 2004 Spring Rich Transcription Evaluation Workshop*, Montreal, 2004
28. M. Cettolo, M. Vescovi, Efficient audio segmentation algorithms based on the BIC, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, vol. 6 (2003)
29. S. Chen, P. Gopalakrishnan, Speaker, environment and channel change detection and clustering via the Bayesian information criterion, in *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop* (1998), pp. 127–132
30. T. Cover, J. Thomas, *Elements of Information Theory* (Wiley-Interscience, London, 2006)
31. S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust. Speech Signal Process. **28**(4), 357–366 (1980) [see also IEEE Transactions on Signal Processing]
32. N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2011). doi:10.1109/TASL.2010.2064307
33. P. Delacourt, D. Kryze, C. Wellekens, Detection of speaker changes in an audio document, in *Sixth European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 1999)
34. P. Delacourt, C. Wellekens, DISTBIC: a speaker-based segmentation for audio data indexing. Speech Commun. **32**(1–2), 111–126 (2000)
35. A. Dempster, N. Laird, D. Rubin et al., Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. **39**(1), 1–38 (1977)
36. R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification* (Wiley, London, 2012)
37. C. Eckart, Optimal rectifier systems for the detection of steady signals, Scripps Institution of Oceanography, (UC San Diego 1952). Retrieved from: http://escholarship.org/uc/item/3676p6rt
38. E. El-Khoury, C. Senac, R. Andre-Obrecht, Speaker diarization: towards a more robust and portable system, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007 (ICASSP 2007)*, vol. 4 (2007), pp. 489–492. doi:10.1109/ICASSP.2007.366956
39. D.P. Ellis, J.C. Liu, Speaker turn segmentation based on between-channel differences, in *NIST ICASSP 2004 Meeting Recognition Workshop*, Montreal, 2004, pp. 112–117
40. T. Ferguson, A Bayesian analysis of some nonparametric problems. Ann. Stat. **1**(2) 209–230 (1973)
41. J.G. Fiscus, J. Ajot, J.S. Garofolo, The rich transcription 2007 meeting recognition evaluation, in *Multimodal Technologies for Perception of Humans* (Springer, Berlin, 2008), pp. 373–389
42. J.G. Fiscus, J. Ajot, M. Michel, J.S. Garofolo, *The Rich Transcription 2006 Spring Meeting Recognition Evaluation* (Springer, Berlin, 2006)
43. J.G. Fiscus, N. Radde, J.S. Garofolo, A. Le, J. Ajot, C. Laprun, The rich transcription 2005 spring meeting recognition evaluation, in *Machine Learning for Multimodal Interaction* (Springer, Berlin, 2006), pp. 369–389

44. E. Fox, E. Sudderth, M. Jordan, A. Willsky, An HDP-HMM for systems with state persistence, in *Proceedings of the 25th International Conference on Machine Learning* (ACM, New York, 2008), pp. 312–319
45. E.B. Fox, E.B. Sudderth, M.I. Jordan, A.S. Willsky, A sticky HDP-HMM with application to speaker diarization. Ann. Appl. Stat. **5**(2A), 1020–1056 (2011)
46. A. Friedland, B. Vinyals, C. Huang, D. Muller, Fusing short term and long term features for improved speaker diarization, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009)* (2009), pp. 4077–4080. doi:10.1109/ICASSP.2009.4960524
47. G. Friedland, A. Janin, D. Imseng, X. Anguera Miro, L. Gottlieb, M. Huijbregts, M. Knox, O. Vinyals, The ICSI RT-09 speaker diarization system. IEEE Trans. Audio Speech Lang. Process. **20**(2), 371–381 (2012). doi:10.1109/TASL.2011.2158419
48. G. Friedland, O. Vinyals, Y. Huang, C. Muller, Prosodic and other long-term features for speaker diarization. IEEE Trans. Audio Speech Lang. Process. **17**(5), 985–993 (2009). doi:10.1109/TASL.2009.2015089
49. R. Gangadharaiah, B. Narayanaswamy, N. Balakrishnan, A novel method for two-speaker segmentation, in *Interspeech* (2004)
50. J. Gauvain, C. Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans. Speech Audio Process. **2**(2), 291–298 (1994)
51. J.L. Gauvain, L. Lamel, G. Adda, Partitioning and transcription of broadcast news data, in *ICSLP*, vol. 98 (1998), pp. 1335–1338
52. J.T. Geiger, F. Wallhoff, G. Rigoll, GMM-UBM based open-set online speaker diarization, in *Interspeech* (2010), pp. 2330–2333
53. H. Gish, M.H. Siu, R. Rohlicek, Segregation of speakers for speech recognition and speaker identification, in *International Conference on Acoustics, Speech, and Signal Processing, 1991 (ICASSP-91)* (1991), pp. 873–876
54. T. Hain, S. Johnson, A. Tuerk, P. Woodland, S. Young, Segment generation and clustering in the HTK broadcast news transcription system, in *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, vol. 1998 (1998)
55. J. Hansen, B. Zhou, M. Akbacak, R. Sarikaya, B. Pellom, Audio stream phrase recognition for a national gallery of the spoken word:“ One Small Step”, in *Sixth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 2000)
56. H. Hermansky, Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. **87**(4), 1738–1752 (1990)
57. H. Hermansky, N. Morgan, A. Bayya, P. Kohn, RASTA-PLP speech analysis technique, in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1992 (ICASSP-92)*, vol. 1 (1992), pp. 121–124
58. M. Huijbregts, R. Ordelman, F. de Jong, Annotation of heterogeneous multimedia content using automatic speech recognition. Lecture Notes in Computer Science Semantic Multimedia, vol. 4816, (Springer Berlin Heldeberg 2007), pp. 78–90
59. D. Imseng, G. Friedland, An adaptive initialization method for speaker diarization based on prosodic features, in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (2010), pp. 4946–4949
60. D. Istrate, C. Fredouille, S. Meignier, L. Besacier, J.F. Bonastre, NIST RT’05S evaluation: pre-processing techniques and speaker diarization on multiple microphone meetings, in *Machine Learning for Multimodal Interaction* (Springer, Berlin, 2006), pp. 428–439
61. H. Jin, F. Kubala, R. Schwartz, Automatic speaker clustering, in *Proceedings of the DARPA Speech Recognition Workshop* (1997), pp. 108–111
62. Q. Jin, T. Schultz, Speaker segmentation and clustering in meetings, in *Interspeech*, vol. 4 (2004), pp. 597–600
63. S. Johnson, Who spoke when?-automatic segmentation and clustering for determining speaker turns, in *Sixth European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 1999)

64. S.E. Johnson, J. Woodland, Speaker clustering using direct maximisation of the MLLR-adapted likelihood, in *Proceedings of ICSLP 98* (1998), pp. 1775–1779
65. T. Kemp, M. Schmidt, M. Westphal, A. Waibel, Strategies for automatic segmentation of audio data, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000 (ICASSP'00)*, vol. 3 (2000), pp. 1423–1426
66. P. Kenny, G. Boulianne, P. Dumouchel, Eigenvoice modeling with sparse training data. IEEE Trans. Speech Audio Process. **13**(3), 345–354 (2005). doi:10.1109/TSA.2004.840940
67. H. Kim, D. Ertelt, T. Sikora, Hybrid speaker-based segmentation system using model-level clustering, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1 (2005), pp. 745–748
68. B.E. Kingsbury, N. Morgan, S. Greenberg, Robust speech recognition using the modulation spectrogram. Speech Commun. **25**(1), 117–132 (1998)
69. C. Knapp, G. Carter, The generalized correlation method for estimation of time delay. IEEE Trans. Acoust. Speech Signal Process. **24**(4), 320–327 (1976)
70. T. Koshinaka, K. Nagatomo, K. Shinoda, Online speaker clustering using incremental learning of an ergodic hidden Markov model, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009)* (2009), pp. 4093–4096. doi:10.1109/ICASSP.2009.4960528
71. R. Kuhn, J.C. Junqua, P. Nguyen, N. Niedzielski, Rapid speaker adaptation in eigenvoice space. IEEE Trans. Speech Audio Process. **8**(6), 695–707 (2000)
72. I. Lapidot, SOM as likelihood estimator for speaker clustering, in *Eighth European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 2003)
73. K. Laskowski, C. Fugen, T. Schultz, Simultaneous multispeaker segmentation for automatic meeting recognition, in *Proceedings of EUSIPCO*, Poznan, 2007, pp. 1294–1298
74. K. Laskowski, Q. Jin, T. Schultz, Crosscorrelation-based multispeaker speech activity detection, in *Eighth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 2004)
75. K. Laskowski, G. Karlsruhe, T. Schultz, A geometric interpretation of non-target-normalized maximum cross-channel correlation for vocal activity detection in meetings, in Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, pp. 89–92. Association for Computational Linguistics (2007)
76. K. Laskowski, T. Schultz, Unsupervised learning of overlapped speech model parameters for multichannel speech activity detection in meetings, in *Proceedings of ICASSP* (2006), pp. 993–996
77. V.B. Le, O. Mella, D. Fohr, et al., Speaker diarization using normalized cross likelihood ratio, in *Interspeech*, vol. 7 (2007), pp. 1869–1872
78. D.A. van Leeuwen, The TNO speaker diarization system for NIST RT05s meeting data, in *Machine Learning for Multimodal Interaction* (Springer, Berlin, 2006), pp. 440–449
79. D.A. van Leeuwen, M. Konečný, Progress in the AMIDA speaker diarization system for meeting data, in *Multimodal Technologies for Perception of Humans* (Springer, Berlin, 2008), pp. 475–483
80. D. Lilt, F. Kubala, Online speaker clustering, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004 (ICASSP'04)*, vol. 1 (2004), pp. 333–336
81. D. Liu, F. Kubala, Fast speaker change detection for broadcast news transcription and indexing, in *Sixth European Conference on Speech Communication and Technology* (1999)
82. J. López, D. Ellis, Using acoustic condition clustering to improve acoustic change detection on broadcast news, in *Sixth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 2000)
83. L. Lu, H. Zhang, Real-time unsupervised speaker change detection, in *International Conference on Pattern Recognition*, vol. 16 (2002), pp. 358–361
84. J. Luque, C. Segura, J. Hernando, Clustering initialization based on spatial information for speaker diarization of meetings, in *Interspeech* (2008), pp. 383–386

85. J. Makhoul, Linear prediction: a tutorial review. Proc. IEEE **63**(4), 561–580 (1975)
86. A. Malegaonkar, A. Ariyaeeinia, P. Sivakumaran, J. Fortuna, Unsupervised speaker change detection using probabilistic pattern matching. IEEE Signal Process. Lett. **13**(8), 509–512 (2006)
87. K. Markov, S. Nakamura, Never-ending learning system for on-line speaker diarization, in *IEEE Workshop on Automatic Speech Recognition Understanding, 2007 (ASRU)* (2007), pp. 699–704. doi:10.1109/ASRU.2007.4430197
88. K. Markov, S. Nakamura, Improved novelty detection for online GMM based speaker diarization, in *Interspeech* (2008), pp. 363–366
89. S. Meignier, J. Bonastre, S. Igounet, E-HMM approach for learning and adapting sound models for speaker indexing, in *2001: A Speaker Odyssey-The Speaker Recognition Workshop* (ISCA, Pittsburgh, 2001)
90. S. Meignier, D. Moraru, C. Fredouille, J.F. Bonastre, L. Besacier, Step-by-step and integrated approaches in broadcast news speaker diarization. Comput. Speech Lang. **20**(2–3), 303–330 (2006). doi:http://dx.doi.org/10.1016/j.csl.2005.08.002. http://www.sciencedirect.com/science/article/pii/S0885230805000471
91. X.A. Miró, Robust speaker diarization for meetings, Ph.D. thesis, Universitat Politècnica de Catalunya, Barcelona (2006)
92. D. Moraru, S. Meignier, L. Besacier, J.F. Bonastre, I. Magrin-Chagnolleau, The ELISA consortium approaches in speaker segmentation during the NIST 2002 speaker recognition evaluation, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP'03)*, vol. 2 (2003), p. II-89
93. D. Moraru, S. Meignier, C. Fredouille, L. Besacier, J.F. Bonastre, The ELISA consortium approaches in broadcast news speaker segmentation during the NIST 2003 rich transcription evaluation, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004 (ICASSP'04)*, vol. 1 (2004), p. I-373
94. K. Mori, S. Nakagawa, Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001 (ICASSP'01)*, vol. 1 (2001)
95. R.M. Neal, G.E. Hinton, A view of the em algorithm that justifies incremental, sparse, and other variants, in *Learning in Graphical Models* (Springer, Berlin, 1998), pp. 355–368
96. A.Y. Ng, M.I. Jordan, Y. Weiss et al., On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. **2**, 849–856 (2002)
97. P. Nguyen, L. Rigazio, Y. Moh, J. Junqua, Rich transcription 2002 site report, Panasonic Speech Technology Laboratory (PSTL), in *Proceedings of the 2002 Rich Transcription Workshop* (2002)
98. M. Nishida, T. Kawahara, Unsupervised speaker indexing using speaker model selection based on Bayesian information criterion, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP'03)*, vol. 1 (2003), pp. 172–175
99. J.M. Pardo, X. Anguera, C. Wooters, Speaker diarization for multi-microphone meetings using only between-channel differences, in *Machine Learning for Multimodal Interaction* (Springer, Berlin, 2006), pp. 257–264
100. J.M. Pardo, X. Anguera, C. Wooters, Speaker diarization for multiple distant microphone meetings: mixing acoustic features and inter-channel time differences, in *Interspeech* (2006)
101. J.M. Pardo, R. Barra-Chicote, R. San-Segundo, R. de Córdoba, B. Martínez-González, Speaker diarization features: the UPM contribution to the RT09 evaluation. IEEE Trans. Audio Speech Lang. Process. **20**(2), 426–435 (2012)
102. J. Pelecanos, S. Sridharan, Feature warping for robust speaker verification, in *2001: A Speaker Odyssey-The Speaker Recognition Workshop* (2001)
103. L. Perez-Freire, C. Garcia-Mateo, A multimedia approach for audio segmentation in TV broadcast news, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004 (ICASSP'04)*, vol. 1 (2004)
104. T. Pfau, D. Ellis, A. Stolcke, Multispeaker speech activity detection for the ICSI meeting recorder, in *Proceedings of ASRU*, vol. 1 (2001)

105. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
106. W.M. Rand, Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
107. D. Reynolds, E. Singer, B. Carlson, G. O'Leary, J. McLaughlin, M. Zissman, Blind clustering of speech utterances based on speaker and language characteristics, in *Fifth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 1998)
108. D.A. Reynolds, T.F. Quatieri, R.B. Dunn, Speaker verification using adapted Gaussian mixture models. Digit. Signal Process. **10**(1), 19–41 (2000)
109. D.A. Reynolds, R.C. Rose, Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans. Speech Audio Process. **3**(1), 72–83 (1995)
110. D.A. Reynolds, P. Torres-Carrasquillo, The MIT Lincoln laboratory RT-04F diarization systems: applications to broadcast audio and telephone conversations. Technical Report, DTIC Document (2004)
111. M. Roch, Y. Cheng, Speaker segmentation using the MAP-adapted Bayesian information criterion, in *ODYSSEY04-The Speaker and Language Recognition Workshop* (ISCA, Pittsburgh, 2004)
112. P.R. Roth, Effective measurements using digital signal analysis. IEEE Spectr. **8**(4), 62–70 (1971)
113. J. Rougui, M. Rziza, D. Aboutajdine, M. Gelgon, J. Martinez, F. Rabat, Fast incremental clustering of gaussian mixture speaker models for scaling up retrieval in on-line broadcast, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2006 (ICASSP 2006)*, vol. 5 (2006)
114. M. Rouvier, S. Meignier, A global optimization framework for speaker diarization, in *Odyssey 2012-The Speaker and Language Recognition Workshop* (2012)
115. M.A. Sato, S. Ishii, On-line EM algorithm for the normalized Gaussian network. Neural Comput. **12**(2), 407–432 (2000)
116. G. Schwarz, Estimating the dimension of a model. Ann. Stat. **6**(2), 461–464 (1978)
117. E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, A. Stolcke, Modeling prosodic feature sequences for speaker recognition. Speech Commun. **46**(3), 455–472 (2005)
118. S. Shum, N. Dehak, E. Chuangsuwanich, D.A. Reynolds, J.R. Glass, Exploiting intra-conversation variability for speaker diarization, in *Interspeech* (2011), pp. 945–948
119. S. Shum, N. Dehak, R. Dehak, J. Glass, Unsupervised methods for speaker diarization: an integrated and iterative approach. IEEE Trans. Audio Speech Lang. Process. **21**(10), 2015–2028 (2013). doi:10.1109/TASL.2013.2264673
120. S. Shum, N. Dehak, J. Glass, On the use of spectral and iterative methods for speaker diarization. System **1**(w2), 2 (2012)
121. M.A. Siegler, U. Jain, B. Raj, R.M. Stern, Automatic segmentation, classification and clustering of broadcast news audio, in *Proceedings of DARPA Broadcast News Workshop* (1997), p. 11
122. J. Silovsky, J. Prazak, Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 4193–4196
123. R. Sinha, S.E. Tranter, M.J. Gales, P.C. Woodland, The Cambridge university March 2005 speaker diarisation system, in *Interspeech* (2005), pp. 2437–2440
124. P. Sivakumaran, J. Fortuna, A.M. Ariyaeeinia, On the use of the Bayesian information criterion in multiple speaker detection, in *Interspeech* (2001), pp. 795–798
125. A. Solomonoff, A. Mielke, M. Schmidt, H. Gish, Clustering speakers by their voices, in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2 (1998), pp. 757–760
126. S. Stevens, J. Volkmann, The relation of pitch to frequency: a revised scale. Am. J. Psychol. **53**(3), 329–353 (1940)

127. H. Sun, B. Ma, S. Kalayar Khine, H. Li, Speaker diarization system for RT07 and RT09 meeting room audio, in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (2010), pp. 4982–4985
128. H. Tang, S. Chu, M. Hasegawa-Johnson, T. Huang, Partially supervised speaker clustering. IEEE Trans. Pattern Anal. Mach. Intell. **34**(5), 959–971 (2012). doi:10.1109/TPAMI.2011.174
129. Y. Teh, M. Jordan, M. Beal, D. Blei, Hierarchical Dirichlet processes. J. Am. Stat. Assoc. **101**(476), 1566–1581 (2006)
130. S. Tranter, Two-way cluster voting to improve speaker diarisation performance, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005 (ICASSP'05)*, vol. 1 (2005)
131. A. Tritschler, R. Gopinath, Improved speaker segmentation and segments clustering using the Bayesian information criterion, in *Sixth European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 1999), pp. 679–682
132. W. Tsai, H. Wang, On maximizing the within-cluster homogeneity of speaker voice characteristics for speech utterance clustering, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, 2006
133. W.H. Tsai, S.S. Cheng, Y.H. Chao, H.M. Wang, Clustering speech utterances by speaker using eigenvoice-motivated vector space models, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005 (ICASSP'05)*, vol. 1 (2005), pp. 725–728
134. W.H. Tsai, S.S. Cheng, H.M. Wang, Speaker clustering of speech utterances using a voice characteristic reference space, in *Eighth International Conference on Spoken Language Processing* (2004)
135. F. Valente, Infinite models for speaker clustering, in *Ninth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 2006)
136. F. Valente, C. Wellekens, Variational Bayesian speaker clustering, in *ODYSSEY04-The Speaker and Language Recognition Workshop* (ISCA, Pittsburgh, 2004)
137. F. Valente, C. Wellekens, Variational Bayesian adaptation for speaker clustering, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005 (ICASSP'05)*, vol. 1 (2005)
138. D. Van Leeuwen, T. Factors, The TNO speaker diarization system for NIST RT05s meeting data. Lecture Notes in Computer Science, Machine Learning for Multimodal Interaction (Springer Berlin Heidelberg 2006) vol. 3869, pp. 440
139. A. Vandecatseye, J. Martens, A fast, accurate and stream-based speaker segmentation and clustering algorithm, in *Eighth European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 2003)
140. D. Vijayasenan, F. Valente, Speaker diarization of meetings based on large TDOA feature vectors, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 4173–4176. doi:10.1109/ICASSP.2012.6288838
141. D. Vijayasenan, F. Valente, H. Bourlard, Agglomerative information bottleneck for speaker diarization of meetings data, in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)* (2007), pp. 250–449
142. D. Vijayasenan, F. Valente, H. Bourlard, Combination of agglomerative and sequential clustering for speaker diarization, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 (ICASSP 2008)* (2008), pp. 4361–4364. doi:10.1109/ICASSP.2008.4518621
143. D. Vijayasenan, F. Valente, H. Bourlard, Integration of TDOA features in information bottleneck framework for fast speaker diarization, in *Interspeech* (2008), pp. 40–43
144. D. Vijayasenan, F. Valente, H. Bourlard, Mutual information based channel selection for speaker diarization of meetings data, in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009)* (2009), pp. 4065–4068. doi:10.1109/ICASSP.2009.4960521

145. D. Vijayasenan, F. Valente, H. Bourlard, An information theoretic combination of MFCC and TDOA features for speaker diarization. IEEE Trans. Audio Speech Lang. Process. **19**(2), 431–438 (2011). doi:10.1109/TASL.2010.2048603
146. D. Vijayasenan, F. Valente, H. Bourlard, Multistream speaker diarization of meetings recordings beyond MFCC and TDOA features. Speech Commun. **54**(1), 55–67 (2012)
147. O. Vinyals, G. Friedland, Modulation spectrogram features for improved speaker diarization, in *Interspeech* (2008), pp. 630–633
148. A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory **13**(2), 260–269 (1967)
149. H. Wang, S. Cheng, METRIC-SEQDAC: a hybrid approach for audio segmentation, in *Eighth International Conference on Spoken Language Processing* (ISCA, Pittsburgh, 2004)
150. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*, vol. 8 (MIT Press, Cambridge, 1964)
151. A. Willsky, H. Jones, A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. IEEE Trans. Automat. Contr. **21**(1), 108–112 (1976)
152. C. Wooters, J. Fung, B. Peskin, X. Anguera, Towards robust speaker segmentation: the ICSI-SRI fall 2004 diarization system, in *RT-04F Workshop*, vol. 23 (2004)
153. C. Wooters, M. Huijbregts, The ICSI RT07s speaker diarization system, in *Multimodal Technologies for Perception of Humans* (Springer, Berlin, 2008), pp. 509–519
154. S. Wrigley, G. Brown, V. Wan, S. Renals, Feature selection for the classification of crosstalk in multi-channel audio, in *Eighth European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 2003)
155. S. Wrigley, G. Brown, V. Wan, S. Renals, Speech and crosstalk detection in multichannel audio. IEEE Trans. Speech Audio Process. **13**(1), 84–91 (2005)
156. T. Wu, L. Lu, K. Chen, H. Zhang, UBM-based real-time speaker segmentation for broadcasting news, in *ICME 2003*, vol. 2 (2003), pp. 721–724
157. K. Yamanishi, J.I. Takeuchi, G. Williams, P. Milne, On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2000), pp. 320–324
158. M. Zamalloa, L.J. Rodríguez-Fuentes, G. Bordel, M. Penagarikano, J.P. Uribe, Low-latency online speaker tracking on the AMI corpus of meeting conversations, in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (2010), pp. 4962–4965
159. B. Zhou, J. Hansen, Efficient audio stream segmentation via the combined T^2 statistic and Bayesian information criterion. IEEE Trans. Speech Audio Process. **13**(4), 467–474 (2005)
160. B. Zhou, J.H. Hansen, Unsupervised audio stream segmentation and clustering via the Bayesian information criterion, in *Interspeech* (2000), pp. 714–717
161. X. Zhu, C. Barras, L. Lamel, J.L. Gauvain, Speaker diarization: from broadcast news to lectures, in *Machine Learning for Multimodal Interaction* (Springer, Berlin, 2006), pp. 396–406
162. X. Zhu, C. Barras, S. Meignier, J.L. Gauvain, Combining speaker identification and BIC for speaker diarization, in *Interspeech*, vol. 5 (2005), pp. 2441–2444
163. P. Zochova, V. Radova, Modified DISTBIC algorithm for speaker change detection, in *Ninth European Conference on Speech Communication and Technology* (ISCA, Pittsburgh, 2005)
164. E. Zwicker, E. Terhardt, Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. J. Acoust. Soc. Am. **68**, 1523 (1980)

Part III
Current Trends in Speech Enhancement

Chapter 9
Maximum A Posteriori Spectral Estimation with Source Log-Spectral Priors for Multichannel Speech Enhancement

Yasuaki Iwata, Tomohiro Nakatani, Takuya Yoshioka, Masakiyo Fujimoto, and Hirofumi Saito

Abstract When speech signals are captured in real acoustical environments, the captured signals are distorted by certain types of interference, such as ambient noise, reverberation, and extraneous speakers' utterances. There are two important approaches to speech enhancement that reduce such interference in the captured signals. One approach is based on the spatial features of the signals, such as direction of arrival and acoustic transfer functions, and enhances speech using multichannel audio signal processing. The other approach is based on speech spectral models that represent the probability density function of the speech spectra, and it enhances speech by distinguishing between speech and noise based on the spectral models. In this chapter, we propose a new approach that integrates the above two approaches. The proposed approach uses the spatial and spectral features of signals in a complementary manner to achieve reliable and accurate speech enhancement. The approach can be applied to various speech enhancement problems, including denoising, dereverberation, and blind source separation (BSS).

Y. Iwata
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

NTT Communication Science Laboratories, NTT Corporation, 3-4, Hikaridai, Seikacho, Sorakugun, Kyoto 619-0237, Japan
e-mail: iwata.yasuaki@nttcom.co.jp

T. Nakatani (✉) • T. Yoshioka • M. Fujimoto
NTT Communication Science Laboratories, NTT Corporation, 3-4, Hikaridai, Seikacho, Sorakugun, Kyoto 619-0237, Japan
e-mail: nakatani.tomohiro@lab.ntt.co.jp; yoshioka.takuya@lab.ntt.co.jp; fujimoto.masakiyo@lab.ntt.co.jp

H. Saito
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
e-mail: saito@is.nagoya-u.ac.jp

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_9

In particular, in this chapter, we focus on applying the approach to BSS. We show experimentally that the proposed integration can improve the performance of BSS compared with a conventional approach.

9.1 Introduction

Speech is an important medium of human-human communication, and it can also constitute a useful human-computer interface thanks to recent advances in automatic speech recognition (ASR) techniques. However, speech signals captured by microphones in real acoustical environments usually contain various types of interference, such as ambient noise, reverberation, and extraneous speakers' utterances. Such interference may seriously degrade speech intelligibility and ASR performance, and thus limits the application areas of speech communication and the speech interface.

Speech enhancement is a framework that acoustically enhances the desired speech in the captured signals by suppressing the interference, and it has been extensively studied to overcome the above problems [1, 14]. Many speech enhancement techniques have been proposed for noise reduction (denoising) [10, 15, 19, 20, 24], reverberation suppression (dereverberation) [6, 16, 25, 27], and source separation [4, 13, 21, 26].

Speech enhancement techniques can be categorized into two approaches based on the features of the sounds that they use.

One approach is based on the spatial features of the individual sounds included in the captured signal. The spatial features are composed, for example, of directions-of-arrival (DOAs) of the sounds [13, 26] and the acoustic transfer functions from the locations of the sound sources to the microphones [4, 6, 16, 21]. In many cases, the speech and the interference have different spatial features and can be distinguished from each other based on these differences, thus making speech enhancement possible. To extract the spatial features, we usually use two or more microphones to capture the speech signals and then subject them to multichannel audio signal processing.

The other approach is based on the spectral features of the individual sounds included in the captured signals [5, 7, 15, 17, 20]. A spectral feature of a sound can be represented, for example, by a power spectrum or a log-power spectrum of the sound. The spectral features of a specific sound usually have a unique distribution, which can be modeled by a probability density function (PDF). Such a PDF represents the kind of spectral features that the sound tends to possess. With this approach, a type of spectral distribution is often used, which we refer to as a spectral prior in this chapter. A spectral prior is a distribution of spectral features that is trained in advance using databases of a specific sound. Assuming the spectral features of speech to have a distribution that differs from that of the spectral features of the interference, we can achieve speech enhancement by

distinguishing the speech and the interference based on their spectral features. With an accurate speech spectral prior, this approach would be capable of reliable speech enhancement. To model the various spectral features that the speech spectra can exhibit in different short time frames, Gaussian Mixture Models (GMMs) and/or Hidden Markov Models (HMMs) of the speech log-power spectra are frequently used as the speech spectral priors [5, 15, 20].

The above two approaches have both been shown to achieve good speech enhancement under certain recording conditions, however, they still have some limitations under other conditions. For example, speech enhancement based on spatial features may degrade when more than one sound source is located in the same direction and/or when we cannot use a sufficient number of microphones to distinguish the spatial features of the signals for the problems to be solved. On the other hand, speech enhancement based on spectral features degrades when the speech and the interference have similar spectral distributions, and/or when the spectral prior does not accurately model the distribution of the speech spectra.

To overcome the above limitations, this chapter proposes a new approach that utilizes the above two types of features in a complementary manner. The two types of features reflect different aspects of the speech and the interference, and so even when we have difficulty in distinguishing the speech and the interference using one of the features, we may be able to distinguish them using the other feature. As a consequence, the proposed approach can distinguish between speech and interference under a wide range of recording conditions, and thus can achieve better speech enhancement.

In presenting the proposed approach, we first refer to an existing speech enhancement approach that is based mainly on the spatial features of the signals [4, 10, 16]. The spectral estimation achieved by this approach is called Maximum Likelihood Spectral Estimation (MLSE) in this chapter. With MLSE, we introduce a probabilistic model, referred to as a likelihood function, which represents how the captured signal is generated depending on unknown spectral and spatial features of the speech and the interference. Then, we estimate the spectral and the spatial features of the speech and the interference as those that maximize the likelihood function. MLSE has been applied to a wide range of speech enhancement problems, including denoising [10], dereverberation [16], and source separation [4], and the effectiveness of MLSE has been confirmed for the respective problems. With MLSE, however, we usually use rather simple models for the spectral features, and thus speech enhancement is conducted mainly by finding spatial features that can allow us to distinguish between speech and interference. The use of spectral priors, which could enable us to distinguish between speech and interference, has not been well studied for this approach.

This chapter proposes a versatile framework for introducing spectral priors into MLSE. With the proposed approach, the speech enhancement is accomplished based on Maximum A Posteriori Spectral Estimation (MAPSE). Compared with MLSE, which estimates the speech spectra by maximizing the likelihood function of the captured signal, MAPSE estimates the speech spectra by maximizing the product of

the likelihood function and the spectral priors. As a consequence, MAPSE can take account of the spatial and spectral features simultaneously, thereby achieving more reliable spectral estimation.

Note that we have already employed MAPSE to extend existing speech enhancement techniques, namely denoising [12] and dereverberation [11], and showed its effectiveness in the respective applications. In this chapter, as an additional application of MAPSE, we describe a way of applying it to blind source separation (BSS) and show the effectiveness of the approach.

In the rest of this chapter, Sect. 9.2 first explains the way for representing and modeling signals that is commonly used through this chapter to explain MLSE and MAPSE. Then, after explaining the basic idea behind MLSE in Sect. 9.3, we extend it to MAPSE in Sect. 9.4. Section 9.5 shows the way for applying MAPSE to BSS. The effectiveness of the proposed application is presented in Sect. 9.6 by simulation experiments. Section 9.7 provides our concluding remarks.

9.2 Signal Representation and Modeling for Multichannel Speech Enhancement

In this section, we describe a general speech capture scenario used in this chapter and the way for representing and modeling signals for the scenario. Hereafter, we denote scalar, vector, and matrix variables by lower case symbols with lightface fonts, lower case symbols with boldface fonts, and upper case symbols with boldface fonts, respectively.

9.2.1 General Speech Capture Scenario for Multichannel Speech Enhancement

First, we define an example speech capture scenario, which is illustrated in Fig. 9.1, to explain the idea behind MLSE/MAPSE. In the scenario, a clean speech signal, s_t, is generated by a speaker in a room, where s_t is a digitized sample of the waveform of the clean speech, and t is the index of the sample. Then, a set of acoustic transfer functions (ATFs) first transfer s_t to $z_t^{(m)}$ for individual microphones $m = 1$ to M, respectively. In the scenario, we assume that the impulse responses corresponding to the ATFs are relatively short, e.g., less than 100 ms.[1] Next, the signal $z_t^{(m)}$ is contaminated by interference, and finally captured by a set of M microphones and denoted as captured signal $x_t^{(m)}$.

[1] As noted later, despite this assumption, this scenario can represent a situation with long reverberation, and can be used for achieving dereverberation.

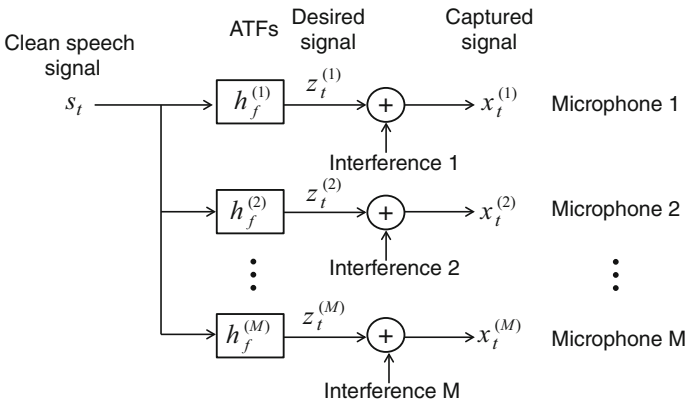

Fig. 9.1 General model of speech signal capture scenario

In this chapter, we assume that the goal of speech enhancement is to reduce the interference contaminated into the captured signal $x_t^{(m)}$, and to estimate $z_t^{(m)}$ for every m. Thus, we refer to $z_t^{(m)}$ as a desired signal.[2]

The above scenario can be viewed as a general scenario that represent various speech capture situations depending on the interpretation. We provide some examples below.

1. The scenario can represent a situation in which we capture a single speaker's utterance in a noisy environment if we interpret the interference in the scenario as ambient noise [10, 12]. Then, the estimation of the desired signal corresponds to the denoising of the captured signal.
2. The scenario can represent a situation in which we capture a single speaker's utterance in a reverberant environment. It is known that the reverberation can be partitioned into the direct sound, early reflections, and late reverberation and that the late reverberation is little correlated with the direct sound and early reflections. Therefore, if we interpret the desired signal, $z_t^{(m)}$, as being composed of the direct sound and the early reflections, and interpret the interference as the late reverberation [11, 16], then the estimation of the desired signal corresponds to the dereverberation of the captured signal.
3. The scenario can represent a situation in which we capture multiple speakers' utterances at the same time if we interpret one of the speaker's utterance as the desired signal and interpret the sum of the other speakers' utterances as the interference [4, 9]. Then, the estimation of the desired signal corresponds to BSS of the captured signal.

[2] If we interpret the ATFs from s_t to $z_t^{(m)}$ also as a part of the interference, we may formulate speech enhancement that estimates s_t. This is beyond the scope of this chapter.

With MLSE/MAPSE based speech enhancement, to estimate the desired signal, we introduce probabilistic models that represents the above speech capture scenario. The models are referred to as generative models. The speech is enhanced by estimating the parameters of the models from the captured signals with the maximum likelihood scheme or with the maximum a posteriori scheme. In the following subsections, after introducing a TF domain representation of the signals that we use for the speech enhancement, we define generative models that represent the above general scenario.

9.2.2 Time-Frequency Domain Representation of Signals

The speech enhancement discussed in this chapter is performed in the TF domain. So, we apply a short-time discrete Fourier transformation (STFT) to the captured signal $x_t^{(m)}$ to obtain its TF representation as

$$x_{n,f}^{(m)} = \sum_{t=0}^{F-1} x_{t+nT_{\text{shift}}}^{(m)} w_t^{\text{ana}} e^{-j\frac{2\pi}{L}ft}, \tag{9.1}$$

where n and f are indices of time frames and frequency bins, F is the number of the discrete Fourier transformation points, w_t^{ana} and T_{shift} are the analysis window and the window shift of the STFT, respectively, and j is the imaginary unit. $x_{n,f}^{(m)}$ for $f = 0, 1, \ldots, F-1$ takes a complex value, and it is called the complex spectrum of the captured signal. The complex spectrum of a clean speech signal and that for each desired signal, $s_{n,f}$ and $z_{n,f}^{(m)}$, are defined in the same way.

With speech enhancement in the TF domain, an estimate of the desired signal is obtained in the TF domain as $\hat{z}_{n,f}^{(m)}$ by applying signal processing to the complex spectrum of the captured signal $x_{n,f}^{(m)}$. Hereafter, a symbol with a hat as in $\hat{z}_{n,f}^{(m)}$ represents an estimated value corresponding to the symbol.

Then, we can obtain an estimate of the desired signal in the time domain, $\hat{z}_t^{(m)}$, by applying an inverse short time discrete Fourier transformation (ISTFT) followed by the overlap-add synthesis as follows:

$$\hat{z}_{n,t}^{(m)} = \frac{1}{L}\sum_{f=0}^{F-1} \hat{z}_{n,f}^{(m)} e^{j\frac{2\pi}{L}ft}, \tag{9.2}$$

$$\hat{z}_t^{(m)} = \sum_{n=1}^{N} w_{t-nT_{\text{shift}}}^{\text{syn}} \hat{z}_{n,t-nT_{\text{shift}}}^{(m)}, \tag{9.3}$$

where N is the number of short time frames and w_t^{syn} is the synthesis window for the overlap-add synthesis.

Hereafter, for the sake of simplicity, we refer to the complex spectrum of a signal simply as a signal without ambiguity unless otherwise noted. For example, we refer

to the complex spectrum of a clean speech signal as a clean speech signal. Further, we use several different vector representations of a signal in this chapter. First, we define three types of vectors of a signal, namely spatial, spectral, and temporal vectors. Letting T indicate a non-conjugate transpose operation, the three types of vectors are defined, for example for $x_{n,f}^{(m)}$, as

$$\begin{aligned} &\text{(Spatial vector)} && \mathbf{x}_{n,f} = [x_{n,f}^{(1)}, x_{n,f}^{(2)}, \ldots, x_{n,f}^{(M)}]^T, \\ &\text{(Spectral vector)} && \mathbf{x}_{n}^{(m)} = [x_{n,0}^{(m)}, x_{n,1}^{(m)}, \ldots, x_{n,F-1}^{(m)}]^T, \\ &\text{(Temporal vector)} && \mathbf{x}_{f}^{(m)} = [x_{1,f}^{(m)}, x_{2,f}^{(m)}, \ldots, x_{N,f}^{(m)}]^T, \end{aligned} \tag{9.4}$$

where the spatial, spectral, and temporal vectors, respectively, contain $x_{n,f}^{(m)}$ for all the microphones $m = 1, \ldots, M$, for all the frequency bins $f = 0, \ldots, F-1$, and for all time frames $n = 1, \ldots, N$. The symbols of the three types of vectors are denoted by using a bold face font and by dropping one of the three indices, m, f, or n, from the original symbol, $x_{n,f}^{(m)}$. We can define the three types of vectors for all other symbols in the TF domain in a similar way. In addition, we use vector representations that combine two or three of the above representations. For example, by cascading spectral vectors $\mathbf{x}_n^{(m)}$ over all the time frames, we can compose a temporal-spectral vector of $x_{n,f}^{(m)}$ for all TF points, which is defined as

$$\text{(Temporal} - \text{spectral vector)} \quad \mathbf{x}^{(m)} = [(\mathbf{x}_{n=1}^{(m)})^T, (\mathbf{x}_{n=2}^{(m)})^T, \ldots, (\mathbf{x}_{n=N}^{(m)})^T]^T, \tag{9.5}$$

where the symbol of the temporal-spectral vector is denoted by dropping both of the frame and frequency indices from $x_{n,f}^{(m)}$. On the right hand side of the above equation we indicate the frame indices of the variables with its index type (e.g., "$n =$" in "$n = 1$") because otherwise the index type is ambiguous. Other combination representations can also be defined in a similar way.

9.2.3 Generative Model of Desired Signals

For MLSE/MAPSE, we introduce a simple generative model for a clean speech signal. Specifically, we assume that the PDF of a clean speech signal, $s_{n,f}$, can be modeled separately at each TF point by

$$p(s_{n,f}|v_{n,f}) = \mathcal{N}_c^{(1)}(s_{n,f}; 0, v_{n,f}), \tag{9.6}$$

where $p(\mathbf{x}|\mathbf{y})$ represents a PDF of $\mathbf{x}$ conditioned on $\mathbf{y}$, and $\mathcal{N}_c^{(p)}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a p-dimensional complex Gaussian PDF with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. The above equation assumes that the clean speech signal at a TF point, $s_{n,f}$, follows a complex Gaussian PDF with a mean 0 and a variance $v_{n,f}$ defined as

$$v_{n,f} = E\{|s_{n,f}|^2\}, \tag{9.7}$$

where $E\{\cdot\}$ represents an expectation function. The above equation also means that $v_{n,f}$ corresponds to the power of the clean speech signal at a TF point.

To model the nonstationarity of the clean speech signal, MLSE/MAPSE assumes that the power of the clean speech $v_{n,f}$ can take any values at different TF points. Thus, Eq. (9.6) models a PDF of a clean speech signal that can potentially take any power pattern in the TF domain.

Here, we define the generative model of the desired signals, $z_{n,f}^{(m)}$. We first assume that the desired signals can be approximated by the product of the clean speech signals and the ATFs in the TF domain as follows:

$$z_{n,f}^{(m)} = h_f^{(m)} s_{n,f}, \tag{9.8}$$

where $h_f^{(m)}$ is the ATF for the m-th microphone in Fig. 9.1. Let $\mathbf{z}_{n,f}$ and $\mathbf{h}_f$ be spatial vectors for the desired signals, $z_{n,f}^{(m)}$, and the ATFs, $h_f^{(m)}$, for all m, respectively. Then, Eq. (9.8) can also be written as

$$\mathbf{z}_{n,f} = \mathbf{h}_f s_{n,f}. \tag{9.9}$$

Further, letting H indicate a conjugate transpose operation, we assume that the time-varying spatial correlation matrix of $\mathbf{z}_{n,f}$, namely $E\{\mathbf{z}_{n,f}\mathbf{z}_{n,f}^H\}$, can be decomposed into

$$E\{\mathbf{z}_{n,f}\mathbf{z}_{n,f}^H\} = E\{|s_{n,f}|^2\mathbf{h}_f\mathbf{h}_f^H\} \tag{9.10}$$

$$= E\{|s_{n,f}|^2\}E\{\mathbf{h}_f\mathbf{h}_f^H\} \tag{9.11}$$

$$= v_{n,f}\mathbf{R}_f, \tag{9.12}$$

where $v_{n,f}$ and $\mathbf{R}_f$ are the time-varying and time-invariant parts of the spatial correlation matrix, and defined, respectively, as in Eq. (9.7) and as

$$\mathbf{R}_f = E\{\mathbf{h}_f\mathbf{h}_f^H\}. \tag{9.13}$$

Hereafter, $\mathbf{R}_f$ is referred to as a normalized spatial correlation matrix, and assumed to be non-singular as in [4] to ensure that the spatial correlation matrix defined as in Eq. (9.12) is non-singular.

Then, based on Eqs. (9.6) and (9.12), the generative model of the desired signal is modeled by an M-dimensional complex Gaussian PDF with a zero mean vector, denoted by $\mathbf{0}$, and a covariance matrix that is equal to the spatial correlation matrix. It is defined as

$$p(\mathbf{z}_{n,f}|v_{n,f}, \mathbf{R}_f) = \mathcal{N}_c^{(M)}(\mathbf{z}_{n,f}; \mathbf{0}, v_{n,f}\mathbf{R}_f). \tag{9.14}$$

In the above equation, $v_{n,f}$ and $\mathbf{R}_f$, respectively, represent the spectral and spatial features of the desired signal, namely the time-varying power of the clean speech signal at a TF point, which is common to all the microphones, and the time-invariant normalized spatial correlation matrix of the desired signal at a frequency bin, which models the diversity of $\mathbf{z}_{n,f}$ over different microphones.

9.2.4 Generative Model of Interference

For MLSE/MAPSE, we also introduce another generative model that represents the way in which the captured speech is contaminated by interference. It is defined as a probabilistic model for the process of generating the interference. The model depends on the speech capture situation under consideration, so in the following we explain the idea of MLSE/MAPSE by using an example case, in which the interference can be modeled by a multivariate complex Gaussian PDF[3] that will be used later in this chapter to apply MLSE to BSS [4].

Let $a_{n,f}^{(m)}$ be the interference signal at a TF point captured by the m-th microphone, and $\mathbf{a}_{n,f}$ be a spatial vector of $a_{n,f}^{(m)}$. Then, assuming that the generative model of the interference signal can be modeled by a multivariate complex Gaussian distribution separately at each TF point, it is defined as

$$p(\mathbf{a}_{n,f}|\theta_f) = \mathcal{N}_c^{(M)}(\mathbf{a}_{n,f}; \mathbf{0}, \mathbf{\Sigma}_{n,f}(\theta_f)), \tag{9.15}$$

where θ_f is a set of parameters of the model, and $\mathbf{\Sigma}_{n,f}(\theta_f)$ is a model of the spatial correlation matrix of the interference signal. For example, as will be explained for the application to BSS, $\mathbf{\Sigma}_{n,f}(\theta_f)$ may be modeled by a sum of the spatial correlation matrices of all the interfering sounds that are parameterized by θ_f.

[3]The same model can be used to represent ambient noise, for example, as in [10]. The way to formulate MLSE for denoising and its extension to MAPSE can be found in [12]. As regards MLSE based dereverberation with the long-term linear prediction approach, the generative model of the interference can be defined in the following form [10, 11, 16].

$$p(\mathbf{a}_{n,f}|\theta_f) = \delta(\mathbf{a}_{n,f} - \mathbf{r}_{n,f}(\theta_f)), \tag{9.16}$$

where $\delta(\cdot)$ is the Dirac delta function, and $\mathbf{r}_{n,f}(\theta_f) = [r_{n,f}^{(1)}(\theta_f), r_{n,f}^{(2)}(\theta_f), \ldots, r_{n,f}^{(M)}(\theta_f)]^T$ is a spatial vector of the interference signal, namely the late reverberation signal. The model parameter set θ_f is composed of the prediction coefficients, and the late reverberation $r_{n,f}^{(m)}(\theta_f)$ is modeled by an inner product of a vector containing the prediction coefficients and that containing a past captured signal in the MLSE based dereverberation. It was shown that the MLSE based dereverberation can be extended to MAPSE based dereverberation as discussed in [11] based on the technique discussed in this chapter.

According to the above model and assuming $\mathbf{x}_{n,f} = \mathbf{z}_{n,f} + \mathbf{a}_{n,f}$, we can derive the conditional PDF of the captured signal given the desired signal as

$$p(\mathbf{x}_{n,f}|\mathbf{z}_{n,f}, \theta_f) = \mathcal{N}_c^{(M)}(\mathbf{x}_{n,f}; \mathbf{z}_{n,f}, \mathbf{\Sigma}_{n,f}(\theta_f)). \tag{9.17}$$

9.3 Speech Enhancement Based on Maximum Likelihood Spectral Estimation (MLSE)

This chapter outlines the way that MLSE estimates speech spectra based on maximum likelihood estimation using the generative models defined in the previous chapter, and explains how it can be applied to multichannel speech enhancement.

9.3.1 Maximum Likelihood Spectral Estimation (MLSE)

In MLSE, a likelihood function is introduced to estimate the speech spectra based on its maximization. Using the above generative models, the likelihood function is defined as a conditional PDF of the captured signal, $\mathbf{x}$, given a set of unknown parameters to be estimated, $\Theta = \{\mathbf{v}, \mathbf{R}, \theta\}$. In Θ, $\mathbf{v}$ is a temporal-spectral vector composed of $v_{n,f}$ for all TF points, $\mathbf{R}$ is a set of normalized spatial correlation matrices, $\mathbf{R}_{n,f}$, for all TF points, and θ is a combination of sets θ_f of the interference generative model for all frequency bins f. Note that $\mathbf{v}$ is composed of the power spectra of the clean speech at all the time frames, which are hereafter referred to simply as clean speech power spectra. Let $\mathbf{x}$ and $\mathbf{z}$ be temporal-spectral-spatial vectors composed, respectively, of $x_{n,f}^{(m)}$ and $z_{n,f}^{(m)}$ for all time frames n, frequency bins f, and microphones m. Then, the likelihood function is defined as

$$\mathcal{L}(\Theta) = p(\mathbf{x}|\mathbf{v}, \mathbf{R}, \theta). \tag{9.18}$$

It can be further rewritten as

$$p(\mathbf{x}|\mathbf{v}, \mathbf{R}, \theta) = \int p(\mathbf{x}|\mathbf{z}, \theta) p(\mathbf{z}|\mathbf{v}, \mathbf{R}) d\mathbf{z}, \tag{9.19}$$

$$= \prod_{n,f} \int p(\mathbf{x}_{n,f}|\mathbf{z}_{n,f}, \theta_f) p(\mathbf{z}_{n,f}|v_{n,f}, \mathbf{R}_f) d\mathbf{z}_{n,f}. \tag{9.20}$$

In Eq. (9.20), $p(\mathbf{z}_{n,f}|v_{n,f}, \mathbf{R}_f)$ and $p(\mathbf{x}_{n,f}|\mathbf{z}_{n,f}, \theta)$ correspond, respectively, to the generative model of the desired signal as in Eq. (9.14) and to the generative model of the interference defined, for example, by Eq. (9.17).

The MLSE approach estimates the clean speech power spectra $\mathbf{v}$ as the part of the parameters that maximizes the likelihood function in Eq. (9.18) as follows:

$$\hat{\Theta} = \arg\max_{\Theta} \mathscr{L}(\Theta). \tag{9.21}$$

Among the estimated parameters in $\hat{\Theta} = \{\hat{\mathbf{v}}, \hat{\mathbf{R}}, \hat{\theta}\}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{R}}$ represent the spectral and spatial features of the desired signal, respectively, and $\hat{\theta}$ represents the spatial and spectral features of the interference.

When we assume that the interference can be modeled by a complex Gaussian distribution as in the previous section, the above likelihood function can be further rewritten as

$$p(\mathbf{x}|\mathbf{v}, \mathbf{R}, \theta) = \prod_{n,f} \mathcal{N}_c^{(M)}(\mathbf{x}_{n,f}; \mathbf{0}, \boldsymbol{\Sigma}_{n,f}^{(\mathbf{x})}), \tag{9.22}$$

where

$$\boldsymbol{\Sigma}_{n,f}^{(\mathbf{x})} = v_{n,f}\mathbf{R}_f + \boldsymbol{\Sigma}_{n,f}(\theta_f). \tag{9.23}$$

Equation (9.22) means that $\mathbf{x}_{n,f}$ has a multivariate complex Gaussian distribution with a zero mean vector and a time-varying spatial correlation matrix, $\boldsymbol{\Sigma}_{n,f}^{(\mathbf{x})}$, which is parameterized by Eq. (9.23). Then, by maximizing the likelihood function, MLSE jointly estimates all the parameters, $\mathbf{v}$, $\mathbf{R}$, and θ, so that the multichannel captured signal, $\mathbf{x}_{n,f}$, best matches the time-varying spatial correlation matrix, $\boldsymbol{\Sigma}_{n,f}^{(\mathbf{x})}$ in Eq. (9.23), based on Eq. (9.21).

A specific procedure for maximizing the likelihood function will be presented for BSS in Sect. 9.5.1.

9.3.2 *Processing Flow of MLSE Based Speech Enhancement*

Figure 9.2 shows the general processing flow of speech enhancement based on MLSE. It is also summarized in Algorithm 1.

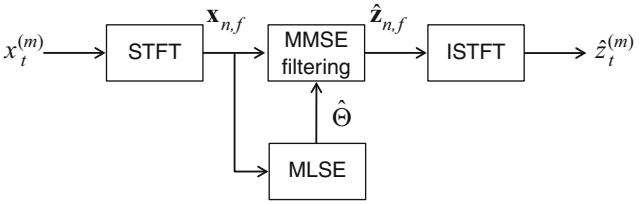

Fig. 9.2 Processing flow of MLSE based speech enhancement

Algorithm 1 Processing flow of MLSE

1. Apply STFT to $x_t^{(m)}$ in the time domain for all m, and obtain the captured signal $\mathbf{x}$ in the TF domain.
2. The clean speech power spectra, $\mathbf{v}$, the normalized spatial correlation matrices, $\mathbf{R}$, and the parameters of the interference generative model, θ, are estimated as $\hat{\mathbf{v}}$, $\hat{\mathbf{R}}$, and $\hat{\theta}$, respectively, based on Eq. (9.21).
3. The desired signal, $\mathbf{z}_{n,f}$, is estimated based on the minimum mean square error (MMSE) estimation [8] using the estimated parameters, $\hat{\mathbf{v}}$, $\hat{\mathbf{R}}$, and $\hat{\theta}$, as follows:

$$\hat{\mathbf{z}}_{n,f} = \int \mathbf{z}_{n,f}\, p(\mathbf{z}_{n,f}|\mathbf{x}_{n,f}, \hat{\mathbf{v}}_{n,f}, \hat{\mathbf{R}}_f, \hat{\theta}_f) d\mathbf{z}_{n,f}, \tag{9.24}$$

When we assume as in Sect. 9.2.4 that the interference can be modeled by a complex Gaussian distribution, the above estimation results in the well-known multichannel Wiener filter, which is defined as

$$\hat{\mathbf{z}}_{n,f} = \left(\frac{1}{\hat{\mathbf{v}}_{n,f}}\hat{\mathbf{R}}_f^{-1} + \boldsymbol{\Sigma}_{n,f}^{-1}(\hat{\theta}_f)\right)^{-1} \frac{1}{v_{n,f}}\hat{\mathbf{R}}_f^{-1}\mathbf{x}_{n,f}. \tag{9.25}$$

4. Apply ISTFT to $\hat{\mathbf{z}}$ to obtain $\hat{z}_t^{(m)}$ in the time domain for all m.

9.4 Speech Enhancement Based on Maximum A Posteriori Spectral Estimation (MAPSE)

While MLSE estimates clean speech power spectra by maximizing the likelihood function, which is defined as the conditional PDF of a captured signal given clean speech power spectra, MAPSE estimates the clean speech power spectra by maximizing the Maximum A Posteriori (MAP) function, which is defined as the conditional PDF of clean speech power spectra given a captured signal. It is defined and rewritten as follows:

$$\mathscr{M}(\Theta) = p(\mathbf{v}|\mathbf{x}, \mathbf{R}, \theta), \tag{9.26}$$

$$= \frac{p(\mathbf{x}|\mathbf{v}, \mathbf{R}, \theta)\, p(\mathbf{v})}{p(\mathbf{x})}, \tag{9.27}$$

$$\propto p(\mathbf{x}|\mathbf{v}, \mathbf{R}, \theta)\, p(\mathbf{v}), \tag{9.28}$$

where $\Theta = \{\mathbf{v}, \mathbf{R}, \theta\}$ is a set of unknown parameters to be estimated as in the likelihood function for MLSE. As indicated in the above equations, the MAP function is proportional to the product of the likelihood function, $p(\mathbf{x}|\mathbf{v}, \mathbf{R}, \theta)$, for MLSE and the prior distribution of the clean speech power spectra, namely the spectral prior, $p(\mathbf{v})$. Accordingly, MAPSE estimates the speech spectra taking account of both the likelihood function that represents the generative model of the captured signal depending on unknown spectral and spatial features of the speech and the interference, and the speech spectral prior that represents the kind of spectral

features that the speech tends to possess. As a consequence, MAPSE can avoid cases where the estimated speech spectra have values that they could never have based on the spectral prior. In this sense, MAPSE can be more reliable than MLSE.

In MAPSE proposed in this chapter, we model the log-power spectra, $\boldsymbol{\rho}_n$, of a clean speech signal with spectral priors, instead of directly modeling the power spectra, $\mathbf{v}_n$, of the signal. Further, we use a Gaussian Mixture Model (GMM) to represent the spectral prior of the log-power spectra, denoted by $p(\boldsymbol{\rho}_n)$, as will be described in Sect. 9.4.2. As a consequence, MAPSE provides better accuracy than MLSE for spectral estimation based on the spectral prior.

9.4.1 Maximum A Posteriori Spectral Estimation (MAPSE)

While MLSE estimates the power of a clean speech signal, $v_{n,f}$, based on maximum likelihood estimation, MAPSE estimates the log-power of a signal, $\rho_{n,f}$, based on MAP estimation. The relationship between $\rho_{n,f}$ and $v_{n,f}$ is defined as

$$\rho_{n,f} = \log v_{n,f}. \tag{9.29}$$

Let $\boldsymbol{\rho}$ be a temporal-spectral vector of $\rho_{n,f}$ for all n and f, which is referred to as the log-power spectra, hereafter. Then, based on Eq. (9.29), we re-define the MAP function in Eq. (9.27) as

$$\mathscr{M}(\Theta) = p(\mathbf{x}, \boldsymbol{\rho}|\mathbf{R}, \theta), \tag{9.30}$$

$$= p(\mathbf{x}|\boldsymbol{\rho}, \mathbf{R}, \theta)p(\boldsymbol{\rho}), \tag{9.31}$$

where we omitted a constant term $p(\mathbf{x})$ because it does not depend on $\boldsymbol{\rho}$, $\mathbf{R}$ or θ, and we set $\Theta = \{\boldsymbol{\rho}, \mathbf{R}, \theta\}$. $p(\mathbf{x}|\boldsymbol{\rho}, \mathbf{R}, \theta)$ in the above equation is the likelihood function of $\mathbf{x}$ given the log-power spectra $\boldsymbol{\rho}$, and it is equal to the likelihood function in Eq. (9.18) via the relationship in Eq. (9.29), that is

$$p(\mathbf{x}|\boldsymbol{\rho}, \mathbf{R}, \theta) = p(\mathbf{x}|\mathbf{v} = \exp(\boldsymbol{\rho}), \mathbf{R}, \theta), \tag{9.32}$$

where $\exp(\boldsymbol{\rho})$ is the element-wise exponential function. $p(\boldsymbol{\rho})$ in Eq. (9.31) is the spectral prior for $\boldsymbol{\rho}$. Letting $\boldsymbol{\rho}_n$ be a spectral vector of $\rho_{n,f}$ at a time frame n, we further assume that $p(\boldsymbol{\rho})$ in Eq. (9.31) can be decomposed into $p(\boldsymbol{\rho}) = \prod_n p(\boldsymbol{\rho}_n)$, where $p(\boldsymbol{\rho}_n)$ is the spectral prior for the log-power spectrum $\boldsymbol{\rho}_n$.

Similar to MLSE, MAPSE estimates the clean speech power spectra $\boldsymbol{\rho}$ as the part of the parameters that maximizes the MAP function in Eq. (9.31) as follows:

$$\hat{\Theta} = \arg\max_{\Theta} \mathscr{M}(\Theta). \tag{9.33}$$

9.4.2 *Log-Spectral Prior of Speech*

MAPSE described in this chapter uses a GMM for modeling the spectral prior, $p(\boldsymbol{\rho})$, of speech log-power spectra. This model is hereafter referred to as the Log-Power Spectral GMM (LPS-GMM) [5,15,20]. This section discusses why we use the LPS-GMM as the model for the log-power spectra, and defines it concretely.

In MLSE, a clean speech signal, $\mathbf{v}$, is estimated as the variance of the quasi-stationary Gaussian distribution in Eq. (9.6). With this model, we may adopt the inverse Gamma distribution as the prior distribution of $v_{n,f}$. The inverse Gamma distribution is known as the conjugate prior for the variance of a univariate Gaussian distribution. It allows us to obtain an analytical solution to the maximization of the MAP function [2]. However, the distribution shape that an inverse Gamma distribution can represent is limited to a relatively simple one, and thus it is difficult to accurately model the distribution of the speech log-power spectra using the inverse Gamma distribution. So, we adopt the LPS-GMM, which can more accurately model the distribution of the speech power spectra.

We adopt the log-power spectra instead of the power spectra as the spectral features because an entire set of speech log-power spectra can be clustered into certain subsets, each of which can be well modeled by a Gaussian distribution. Here, we explain this in an intuitive manner using Fig. 9.3. The figure shows histograms of the power of a speech signal at 3 kHz and that of the log power of the speech signal, which correspond to speech short time frames of the utterance /a/. While the histogram of the power has a sharp peak around zero with a heavy tail on the positive half line, the histogram of the log power has a symmetric unimodal shape like a Gaussian distribution. This suggests that the log power could be well modeled by a Gaussian distribution.

The advantage of using the GMMs for the spectral prior can be explained as follows. As indicated in Fig. 9.3, the power of a speech signal corresponding to a

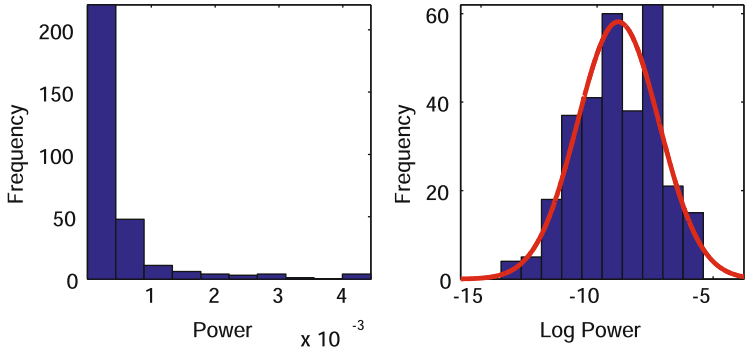

Fig. 9.3 Histograms of power (*left*) and log-power (*right*) at 3 kHz corresponding to speech short time frames of the utterance /a/. A *red solid line* in the *left panel* shows a Gaussian distribution fitted to the histogram

type of phoneme /a/ can be well modeled by a single Gaussian. However, actual speech is composed not only of /a/ but also of many other phonemes, which have spectral shapes that differ greatly from those of /a/. So, the distribution of the speech log power for many phonemes has a multimodal shape that can be viewed as a weighted sum of many Gaussians. Therefore, it is straightforward to use a GMM to model the distribution of the speech log power as a whole. It may be important to note that it is well-known that the use of a GMM is advantageous when modeling the envelopes of speech log-power spectra and for achieving accurate ASR by computer [23].

Now let us define the LPS-GMM used for MAPSE in this chapter. We model the log-power spectrum, $\boldsymbol{\rho}_n$, at a time frame n by a GMM as follows.

$$p(\boldsymbol{\rho}_n) = \sum_{k=1}^{K} p(\boldsymbol{\rho}_n, k_n = k), \tag{9.34}$$

$$= \sum_{k=1}^{K} \lambda_k \, p(\boldsymbol{\rho}_n | k_n = k), \tag{9.35}$$

$$p(\boldsymbol{\rho}_n | k_n = k) = \mathcal{N}^{(L)}(\boldsymbol{\rho}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{9.36}$$

where $\mathcal{N}^{(p)}(\cdot)$ is a PDF of a p-dimensional real-valued Gaussian distribution, K is the number of Gaussians of the GMM, and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are mean vector and covariance matrix of the k-th Gaussian, respectively. In the above equation, we assume that one of the Gaussians defined in Eq. (9.36) is activated at each time frame n, and generates a log-power spectra $\boldsymbol{\rho}_n$. k_n is the index of the Gaussian that is activated at the time frame n, and is assumed not to be observed, namely dealt with as a hidden variable as in Eq. (9.34). $\lambda_k = p(k_n)$ is the prior distribution of k_n, referred to as the mixture weight of the k-th Gaussian, and is assumed to satisfy

$$\sum_{k=1}^{K} \lambda_k = 1, \quad \lambda_k \geq 0. \tag{9.37}$$

The mean vector $\boldsymbol{\mu}_k$ is composed of mean values in all the frequency bins as

$$\boldsymbol{\mu}_k = [\mu_{k,0}, \mu_{k,2}, \ldots, \mu_{k,F-1}]^T, \tag{9.38}$$

and we assume $\boldsymbol{\Sigma}_k$ to be a diagonal covariance matrix that can be defined as

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} \sigma_{k,0}^2 & & & 0 \\ & \sigma_{k,1}^2 & & \\ & & \ddots & \\ 0 & & & \sigma_{k,F-1}^2 \end{bmatrix}. \tag{9.39}$$

Note that it is desirable to use a full covariance matrix for $\mathbf{\Sigma}_k$ to accurately model the statistical characteristics of the speech log-power spectra, however, the use of the full covariance matrix significantly increases the computational complexity of MAPSE. So, for the sake of computational efficiency, MAPSE adopts a diagonal covariance matrix for $\mathbf{\Sigma}_k$, that is, Eq. (9.36) can be further decomposed as

$$p(\boldsymbol{\rho}_n|k_n = k) = \prod_f p(\rho_{n,f}|k_n = k), \tag{9.40}$$

$$p(\rho_{n,f}|k_n = k) = \mathcal{N}^{(1)}(\rho_{n,f}; \mu_{k,f}, \sigma^2_{k,f}). \tag{9.41}$$

9.4.3 Expectation Maximization (EM) Algorithm

It is difficult to obtain an analytical solution that maximizes the MAP function in Eq. (9.31) because of the use of spectral priors based on GMMs that includes a hidden variable k_n. Instead, MAPSE solves this problem by using an iterative optimization framework, namely the Expectation Maximization (EM) algorithm [3]. Letting X, Y, and Z be observed data, parameters to be estimated, and hidden variables, respectively, the EM algorithm maximizes a MAP function defined as

$$\mathcal{M}(Y) = p(X, Y), \tag{9.42}$$

$$= \int p(X, Y, Z)dZ, \tag{9.43}$$

by alternately iterating the E-step and the M-step. The E-step calculates a function, referred to as the Q-function, based on the updated estimation of Y, denoted by $\hat{Y}$, and the M-step updates $\hat{Y}$ as Y that maximizes the Q-function. The Q-function is defined as

$$Q(Y|\hat{Y}) = E\left\{\log p(X, Y, Z)|\hat{Y}\right\}, \tag{9.44}$$

$$= \sum_Z p(Z|X, \hat{Y}) \log p(X, Y, Z) \tag{9.45}$$

where $E\{\log p(X, Y, Z)|\hat{Y}\}$ is a posterior expectation of $\log p(X, Y, Z)$ given $\hat{Y}$, and is defined as Eq. (9.45).

With MAPSE, a set of parameters to be estimated is summarized as $\Theta = \{\boldsymbol{\rho}, \mathbf{R}, \theta\}$. With the LPS-GMM, the hidden variable is the index of the Gaussians that are activated at each time frame n, namely k_n, in Eq. (9.36). Letting $\mathbf{k}$ be a set of k_n for all n, the Q-function can be defined and rewritten as

$$Q(\Theta|\hat{\Theta}) = E\{\log p(\mathbf{x}, \boldsymbol{\rho}, \mathbf{k}|\mathbf{R}, \theta)|\hat{\Theta}\}, \tag{9.46}$$

$$= \log p(\mathbf{x}|\boldsymbol{\rho}, \mathbf{R}, \theta) + E\{\log p(\boldsymbol{\rho}, \mathbf{k})|\hat{\boldsymbol{\rho}}\}. \tag{9.47}$$

Each term in the above equation can further be rewritten as

$$\log p(\mathbf{x}|\boldsymbol{\rho}, \mathbf{R}, \theta) = \sum_{n=1}^{N} \sum_{f=0}^{F-1} \log p(\mathbf{x}_{n,f}|\rho_{n,f}, \mathbf{R}_f, \theta_f), \tag{9.48}$$

$$E\{\log p(\boldsymbol{\rho}, \mathbf{k})|\hat{\boldsymbol{\rho}}\} = \sum_{n=1}^{N} E\{\log p(\boldsymbol{\rho}_n, k_n)|\hat{\boldsymbol{\rho}}_n\}, \tag{9.49}$$

$$= \sum_{n=1}^{N} \sum_{k_n=1}^{K} p(k_n|\hat{\boldsymbol{\rho}}_n) \log p(\boldsymbol{\rho}_n, k_n), \tag{9.50}$$

where

$$\log p(\boldsymbol{\rho}_n, k_n) = \log \lambda_{k_n} + \sum_{f=0}^{F-1} \log p(\rho_{n,f}|k_n). \tag{9.51}$$

As in the above, most of the terms in the Q-function can be decomposed into terms in individual TF points. Therefore, letting Θ_f be a subset of Θ containing the parameters for the generative model at a frequency bin f, the Q-function can be rewritten further as

$$Q(\Theta|\hat{\Theta}) = \sum_{f=0}^{F-1} \sum_{n=1}^{N} Q_{n,f}(\Theta_f|\hat{\Theta}) + \sum_{n=1}^{N} \sum_{k_n=1}^{K} p(k_n|\hat{\boldsymbol{\rho}}_n) \log \lambda_{k_n}, \tag{9.52}$$

$$Q_{n,f}(\Theta_f|\hat{\Theta}) = p(\mathbf{x}_{n,f}|\rho_{n,f}, \mathbf{R}_f, \theta_f) + \sum_{k_n=1}^{K} p(k_n|\hat{\boldsymbol{\rho}}_n) \log p(\rho_{n,f}|k_n). \tag{9.53}$$

In the above equations, $p(k_n|\hat{\boldsymbol{\rho}}_n)$ is a posterior distribution of k_n when $\boldsymbol{\rho}_n = \hat{\boldsymbol{\rho}}_n$ is given, and it can be calculated as

$$p(k_n|\hat{\boldsymbol{\rho}}_n) = \frac{\lambda_{k_n} \mathcal{N}^{(L)}(\hat{\boldsymbol{\rho}}_n; \boldsymbol{\mu}_{k_n}, \boldsymbol{\Sigma}_{k_n})}{\sum_{k_n=1}^{K} \lambda_{k_n} \mathcal{N}^{(L)}(\hat{\boldsymbol{\rho}}_n; \boldsymbol{\mu}_{k_n}, \boldsymbol{\Sigma}_{k_n})}. \tag{9.54}$$

According to the EM algorithm and the above Q-function, we can estimate the log-power spectra, $\boldsymbol{\rho}$, that maximizes the MAP function in Eq. (9.31) via the following steps.

1. Initialize $\hat{\boldsymbol{\rho}}$, $\hat{\mathbf{R}}$, and $\hat{\theta}$.
2. (E-step) Calculate $p(k_n|\hat{\boldsymbol{\rho}}_n)$ for all n with Eq. (9.54).

3. (M-step) Update $\hat{\boldsymbol{\rho}}$, $\hat{\mathbf{R}}$, and $\hat{\theta}$ as $\boldsymbol{\rho}$, $\mathbf{R}$, and θ that maximize (or at least increase) the Q-function defined by Eqs. (9.52) and (9.53). For example, this step may be accomplished by alternate update of $\hat{\boldsymbol{\rho}}$, $\hat{\mathbf{R}}$, and $\hat{\theta}$ as follows:
   a. (M-step1) Assuming $\mathbf{R} = \hat{\mathbf{R}}$ and $\theta = \hat{\theta}$ to be fixed, update $\hat{\rho}_{n,f}$ at each TF point (n, f) as $\rho_{n,f}$ that maximizes Eq. (9.53).
   b. (M-step2) Assuming $\boldsymbol{\rho} = \hat{\boldsymbol{\rho}}$ and $\theta = \hat{\theta}$ to be fixed, update $\hat{\mathbf{R}}_f$ at each TF point (n, f) as $\mathbf{R}_f$ that maximizes the first term in Eq. (9.53).
   c. (M-step3) Assuming $\boldsymbol{\rho} = \hat{\boldsymbol{\rho}}$ and $\mathbf{R} = \hat{\mathbf{R}}$ to be fixed, update $\hat{\theta}_f$ at each frequency bin f as θ_f that maximizes the first term in Eq. (9.52).
4. Iterate the above steps 2 and 3 until convergence is obtained.

A concrete steps for maximizing the MAP function based on the EM algorithm will be presented for BSS in Sect. 9.5.2.2.

9.4.4 Update of $\hat{\boldsymbol{\rho}}_{n,f}$ Based on Newton–Raphson Method

In the above EM iterations, all the parameters that maximize the Q-function in the M-step may be obtained analytically except for the log-power spectra, $\hat{\boldsymbol{\rho}}$. In contrast, $\boldsymbol{\rho}$ that maximizes Eq. (9.53) in M-step1 cannot be obtained analytically because of the non-linear relationship derived from Eq. (9.29) between the likelihood function and the spectral prior. So, it is very important to find an efficient way to update $\hat{\boldsymbol{\rho}}$ for MAPSE. We will explain how we can solve this problem below.

Set the first derivative of Eq. (9.53) that should be zero as $\partial Q_{n,f}/\partial \rho_{n,f}=0$. Based on the generative models given in this chapter for the general speech capture scenario with a certain amount of mathematical manipulations, the resultant equation can be rewritten in the following form,

$$\exp(u) + u + \beta = 0, \tag{9.55}$$

where β is a scalar constant derived from $\partial Q_{n,f}/\partial \rho_{n,f}$ and u is a scalar variable depending only on $\rho_{n,f}$. A concrete derivation of the above equation for BSS will be shown in Sect. 9.5.2.2 and that for dereverberation and denoising can be found in [11, 12]. $\rho_{n,f}$ that maximizes Eq. (9.53) can be obtained by finding u that satisfies the above equation. Since the above equation includes a linear term and a non-linear term, u and $\exp(u)$, we still cannot obtain an analytical solution for it. Instead, we can obtain the solution in a computationally efficient manner by employing the Newton–Raphson method because the left hand side is a monotonically increasing convex function with a scalar variable.

Based on the Newton–Raphson method, the estimation of u can be accomplished via the following steps.

1. Initialize the estimate of u, denoted by $\hat{u}$, as follows

$$\hat{u} = \begin{cases} \log(-\beta) & (\beta \leq -1/2) \\ -\beta & (\beta > -1/2) \end{cases} \tag{9.56}$$

2. Iterate the following steps until convergence is obtained.
   a. Obtain $\hat{u}'$ as

$$\hat{u}' = \hat{u} - \frac{\exp(\hat{u}) + r + \beta}{\exp(\hat{u}) + 1}. \tag{9.57}$$

   b. Update $\hat{u}$ as $\hat{u} = \hat{u}'$

The above step1 allows us to set the initial value of u at a point relatively close to the solution according to the given form of Eq. (9.55). Our preliminary experiment showed that the above procedure gives a good estimate of u, which is very close to the true solution of Eq. (9.55), after only two iterations in most cases.

9.4.5 *Processing Flow*

Figure 9.4 shows the processing flow of MAPSE based speech enhancement. The difference from the processing flow of MLSE based speech enhancement (see Fig. 9.2) is only in the spectral estimation block. The processing flow is also summarized by Algorithm 2.

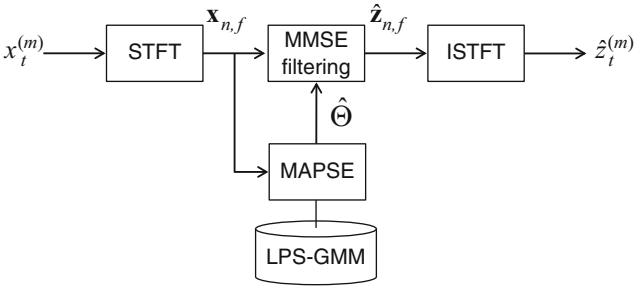

Fig. 9.4 Block diagram of MAPSE based speech enhancement

Algorithm 2 Processing flow of MAPSE

1. Apply STFT to $x_t^{(m)}$ at each time frame for all m, and obtain the captured signal $\mathbf{x}$ in the TF domain.
2. Initialize $\hat{\Theta}$, for example, by applying MLSE to the captured signal with no spectral priors.
3. Estimate $\hat{\Theta}$ that maximizes the MAP function based on MAPSE with a spectral prior modeled by the LPS-GMM.
4. Obtain the desired signal estimate $\hat{\mathbf{z}}$ based on the MMSE estimation based on $\hat{\Theta}$ and $\mathbf{x}$. The MMSE estimation for MAPSE is performed by the same equations as those for MLSE, namely by Eqs. (9.24) and (9.25).
5. Apply ISTFT to $\hat{\mathbf{z}}$ to obtain $z_t^{(m)}$ for all m in the time domain.

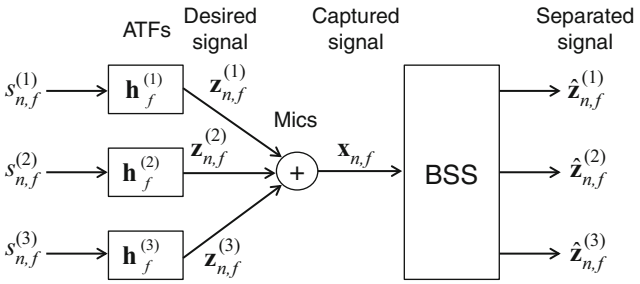

Fig. 9.5 Illustration of speech capture by microphones followed by blind source separation (BSS)

9.5 Application to Blind Source Separation (BSS)

This section describes how we can apply MAPSE to BSS. The goal of BSS is to separate a captured signal, which is a sound mixture composed of more than one sound, into individual sounds. Figure 9.5 is a schematic diagram showing how a sound mixture is captured by microphones and separated into individual sounds by BSS. In particular, when the number of sources exceeds the number of microphones, the BSS framework is referred to as underdetermined BSS. An MLSE based BSS approach has been proposed for underdetermined BSS [4], and it is known as one of the most advanced underdetermined BSS techniques. We refer to it as ML-BSS in this chapter. In the following, we first describe ML-BSS, and then extend it to the MAPSE based BSS approach, referred to as MAP-BSS hereafter, by introducing the LPS-GMMs defined in the previous section as the spectral priors for all the sources.

9.5.1 MLSE for BSS (ML-BSS)

In the following, we first describe ML-BSS before formulating MAP-BSS.

9.5.1.1 Generative Models for ML-BSS

The goal of BSS is to separate a sound mixture into individual sounds. Therefore, the desired signals for BSS are defined for individual sources included in the captured signals. To distinguish the desired signals in equations, hereafter we attach a source index, i, to symbols associated with each source, that is, $z_{n,f}^{(i)}$. Then, we assume that each desired signal is generated based on the generative model that is defined in Eq. (9.14) in Sect. 9.2.3. With the source index, it is rewritten as

$$p(\mathbf{z}_{n,f}^{(i)}|v_{n,f}^{(i)}, \mathbf{R}_f^{(i)}) = \mathcal{N}_c^{(M)}(\mathbf{z}_{n,f}^{(i)}; 0, v_{n,f}^{(i)}\mathbf{R}_f^{(i)}), \quad \text{for } i = 1, 2, \ldots, N_s, \quad (9.58)$$

where N_s is the number of sources. As in [4], we also assume that the normalized spatial correlation matrix, $R_f^{(i)}$, is nonsingular so that the covariance matrix of the above Gaussian PDF is nonsingular.

Then, the captured signal is modeled as the sum of the desired signals as

$$\mathbf{x}_{n,f} = \sum_{i=1}^{N_s} \mathbf{z}_{n,f}^{(i)}. \quad (9.59)$$

Following the general speech capture scenario described in Sect. 9.2.1, the above equation can be interpreted as the process whereby one of the desired signals, $\mathbf{z}_{n,f}^{(i)}$, is contaminated by the interference, which is composed of the sum of the other desired signals. Then, the generative model of the captured signal given a desired signal, which was previously derived as in Eq. (9.17), can be derived for ML-BSS as

$$p(\mathbf{x}_{n,f}|\mathbf{z}_{n,f}^{(i)}, \theta_f^{(i)}) = \mathcal{N}_c^{(M)}(\mathbf{x}_{n,f}; \mathbf{z}_{n,f}^{(i)}, \mathbf{\Sigma}_{n,f}(\theta_f^{(i)})), \quad (9.60)$$

$$\mathbf{\Sigma}_{n,f}(\theta_f^{(i)}) = \sum_{i' \neq i} v_{n,f}^{(i')}\mathbf{R}_{n,f}^{(i')}, \quad (9.61)$$

where $\theta_f^{(i)}$ is a set of parameters for the interference generative model, composed of $v_{n,f}^{(i')}$ and $\mathbf{R}_{n,f}^{(i')}$ for $i' \neq i$ at a frequency bin f, and $\mathbf{\Sigma}_{n,f}(\theta_f^{(i)})$ is the spatial correlation matrix of the interference. Similarly, the generative model of the captured signal given clean speech signals for all the sources, previously derived as in Eq. (9.22), can be derived for ML-BSS as

$$p(\mathbf{x}|\mathbf{v}, \mathbf{R}) = \prod_{n,f} \mathcal{N}_c^{(M)}(\mathbf{x}_{n,f}; \mathbf{0}, \mathbf{R}_{n,f}^{(\mathbf{x})}), \quad (9.62)$$

$$\mathbf{R}_{n,f}^{(\mathbf{x})} = \sum_{i=1}^{N_s} v_{n,f}^{(i)}\mathbf{R}_f^{(i)}, \quad (9.63)$$

where $\mathbf{v}$ and $\mathbf{R}$ are a set of $v_{n,f}^{(i)}$ and that of $\mathbf{R}_{n,f}^{(i)}$ for all TF points and for all sources, and $\mathbf{R}_{n,f}^{(\mathbf{x})}$ is a time-varying spatial correlation matrix of $\mathbf{x}$, which is modeled by the sum of the time-varying spatial correlation matrices of all the sources as in Eq. (9.63).

9.5.1.2 MLSE Based on EM Algorithm

Letting $\Theta = \{\mathbf{v}, \mathbf{R}\}$ be the parameters to be estimated, the likelihood function for the ML-BSS is defined as follows:

$$\mathscr{L}(\Theta) = p(\mathbf{x}|\mathbf{v}, \mathbf{R}). \tag{9.64}$$

Then, MLSE is achieved by obtaining Θ that maximizes the likelihood function as

$$\hat{\Theta} = \arg\max_{\Theta} \mathscr{L}(\Theta). \tag{9.65}$$

Unfortunately, we do not have a closed form solution for the above maximization. Instead, for the maximization, we decompose the parameter set Θ into its subsets, $\Theta^{(i)}$, for $i = 1, \ldots, N_s$, so that each subset is composed of $v_{n,f}^{(i)}$ and $\mathbf{R}_{n,f}^{(i)}$ of a source i for all the TF points, and adopt an iterative optimization scheme, where $\Theta^{(i)}$ for each source i is alternately updated by fixing $\Theta^{(i')}$ for $i' \neq i$. In concrete, as in the following equation, we update $\Theta^{(i)}$ alternately for each i by fixing $\Theta^{(i')}$ for $i' \neq i$ at their previously updated values, denoted by $\hat{\Theta}^{(i')}$.

$$\hat{\Theta}^{(i)} = \arg\max_{\Theta^{(i)}} \mathscr{L}(\hat{\Theta}^{(1)}, \ldots, \hat{\Theta}^{(i-1)}, \Theta^{(i)}, \hat{\Theta}^{(i+1)}, \ldots, \hat{\Theta}^{(N_s)}). \tag{9.66}$$

By iterating the above alternate updates until convergence is obtained, we can update all the parameters in Θ that (locally) maximize the likelihood function.

For the update in Eq. (9.66), we can use the EM algorithm. Dealing with the desired signal, $\mathbf{z}^{(i)}$, as hidden variables, and omitting constant terms, the Q-function for the EM algorithm can be defined and rewritten as

$$Q^{(i)}(\Theta^{(i)}|\hat{\Theta}) = E\{\log p(\mathbf{x}, \mathbf{z}^{(i)}|\mathbf{v}^{(i)}, \mathbf{R}^{(i)}, \hat{\theta}^{(i)})|\hat{\Theta}\}, \tag{9.67}$$

$$= \int p(\mathbf{z}^{(i)}|\mathbf{x}, \hat{\mathbf{v}}^{(i)}, \hat{\mathbf{R}}^{(i)}, \hat{\theta}^{(i)}) \log p(\mathbf{x}, \mathbf{z}^{(i)}|\mathbf{v}^{(i)}, \mathbf{R}^{(i)}, \hat{\theta}^{(i)}) d\mathbf{z}^{(i)}, \tag{9.68}$$

$$= \sum_{n,f} \int p(\mathbf{z}_{n,f}^{(i)}|\mathbf{x}_{n,f}, \hat{\mathbf{v}}_{n,f}^{(i)}, \hat{\mathbf{R}}_{n,f}^{(i)}, \hat{\theta}_{f}^{(i)}) \log p(\mathbf{x}_{n,f}, \mathbf{z}_{n,f}^{(i)}|\mathbf{v}_{n,f}^{(i)}, \mathbf{R}_{n,f}^{(i)}, \hat{\theta}_{f}^{(i)}) d\mathbf{z}_{n,f}^{(i)}. \tag{9.69}$$

Then, based on Eqs. (9.58) and (9.60), we obtain

$$Q^{(i)}(\Theta^{(i)}|\hat{\Theta}) = -\sum_{n,f}\left\{\frac{\mathrm{tr}\left(\left(\mathbf{R}_f^{(i)}\right)^{-1}\hat{\mathscr{R}}_{n,f}^{(i)}\right)}{v_{n,f}^{(i)}} + M\log v_{n,f}^{(i)} + \log\det\mathbf{R}_f^{(i)}\right\}, \tag{9.70}$$

where $\mathrm{tr}(\cdot)$ is a trace of a matrix, $\det(\cdot)$ indicates a determinant of a matrix, and $\hat{\mathscr{R}}_{n,f}^{(i)} = E\{\mathbf{z}_{n,f}^{(i)}\mathbf{z}_{n,f}^{(i)\,H}|\mathbf{x},\hat{\Theta}\}$ is the posterior expectation of the desired signal's spatial correlation matrix obtained by the following equations.

$$\mathbf{W}_{n,f}^{(i)} = \hat{v}_{n,f}^{(i)}\hat{\mathbf{R}}_f^{(i)}\left(\hat{\mathbf{R}}_{n,f}^{(\mathbf{x})}\right)^{-1}, \tag{9.71}$$

$$\hat{\mathbf{z}}_{n,f}^{(i)} = \mathbf{W}_{n,f}^{(i)}\mathbf{x}_{n,f}, \tag{9.72}$$

$$\hat{\mathscr{R}}_{n,f}^{(i)} = \hat{\mathbf{z}}_{n,f}^{(i)}\hat{\mathbf{z}}_{n,f}^{(i)\,H} + (\mathbf{I}-\mathbf{W}_{n,f}^{(i)})\hat{v}_{n,f}^{(i)}\hat{\mathbf{R}}^{(i)}, \tag{9.73}$$

where $\mathbf{I}$ is the identity matrix.

Based on the EM algorithm, the update of $\Theta^{(i)}$ can be performed by iterating the E-step and the M-step with the above Q-function. In the E-step, we obtain posterior expectation of the spatial correlation matrix, $\hat{\mathscr{R}}_{n,f}^{(i)} = E\{\mathbf{z}_{n,f}^{(i)}\mathbf{z}_{n,f}^{(i)\,H}|\mathbf{x},\hat{\Theta}\}$, for all TF points by Eqs. (9.71)–(9.73). In this step, the multichannel filter, $\mathbf{W}_{n,f}^{(i)}$, obtained by Eq. (9.71) is identical to a multichannel Wiener filter, and Eq. (9.72) can be viewed as an operation to apply the filter to the spatial vector of the captured signal, $\mathbf{x}_{n,f}$, to update the estimate of the desired signal, $\hat{\mathbf{z}}_{n,f}^{(i)}$. In the M-step, we update $\hat{v}_{n,f}^{(i)}$ and $\hat{\mathbf{R}}_f^{(i)}$ of the source i for all n and f as those that maximize Eq. (9.70). They are obtained as follows:

$$\hat{v}_{n,f}^{(i)} = \frac{1}{M}\mathrm{tr}\left(\left(\hat{\mathbf{R}}_f^{(i)}\right)^{-1}\hat{\mathscr{R}}_{n,f}^{(i)}\right), \tag{9.74}$$

$$\hat{\mathbf{R}}_f^{(i)} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\hat{v}_{n,f}^{(i)}}\hat{\mathscr{R}}_{n,f}^{(i)}. \tag{9.75}$$

Note that the above update equations can be viewed as an operation that decomposes the time-varying spatial correlation matrix $\hat{\mathscr{R}}_{n,f}^{(i)}$ into the time-varying clean speech power, $\hat{v}_{n,f}^{(i)}$, and the time-invariant normalized spatial correlation matrix, $\hat{\mathbf{R}}_f^{(i)}$.

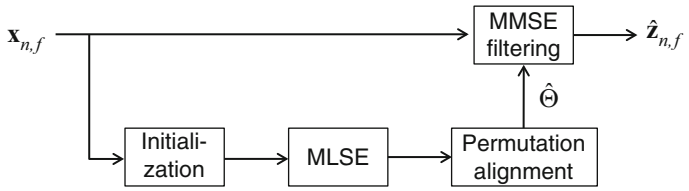

Fig. 9.6 Block diagram of ML-BSS

Algorithm 3 Processing flow of ML-BSS

1. Initialize $\hat{\mathbf{v}}$ and $\hat{\mathbf{R}}$, for example, based on the method proposed in [9].
2. Iterate the following steps until convergence is obtained.
   a. For each source i, perform the following steps.
      i. (E-step) Obtain $\hat{\mathscr{R}}_{n,f}^{(i)}$ based on Eqs. (9.71)–(9.73) for all TF points.
      ii. (M-step) Update $\hat{v}_{n,f}^{(i)}$ and $\hat{\mathbf{R}}_{n,f}^{(i)}$ for all TF points based on Eqs. (9.74) and (9.75).
3. Perform permutation alignment, e.g., based on the method proposed in [21].
4. Perform multichannel Wiener filtering based on Eqs. (9.71) and (9.72) for all i, n, and f, to separate the captured signal, $\mathbf{x}_{n,f}$, into individual desired signals, $\hat{\mathbf{z}}_{n,f}^{(i)}$.

9.5.1.3 Processing Flow of ML-BSS Based on EM Algorithm

Figure 9.6 and Algorithm 3 show the processing flow of ML-BSS based on the EM algorithm.

In the flow, the initialization of $\hat{\mathbf{v}}$ and $\hat{\mathbf{R}}$ is first performed. In this chapter, we assume that the initial values can be set relatively close to the global maximum point, for example, based on a method proposed in [9].

Then, *MLSE* is performed to estimate the parameters, Θ. As written in step 2 of Algorithm 3, the estimation is conducted by applying the EM algorithm to update parameters of each source in turn as in the following.

- In the *E-step of MLSE*, a multichannel filter, $\mathbf{W}_{n,f}^{(i)}$, which is identical to a multichannel Wiener filter, is first calculated by Eq. (9.71) based on updated parameters. Then, $\mathbf{W}_{n,f}^{(i)}$ is applied to the captured signal, $\mathbf{x}_{n,f}$, by Eq. (9.72) to update the estimate of the desired signal, $\hat{\mathbf{z}}_{n,f}^{(i)}$. Finally, the posterior expectation of the spatial correlation matrix for the desired signal, $\hat{\mathscr{R}}_{n,f}^{(i)}$, is updated by Eq. (9.73).
- In the *M-step of MLSE*, the power of the clean speech signal, $\hat{v}_{n,f}^{(i)}$, and the normalized spatial correlation matrix of the desired signal, $\hat{\mathbf{R}}_{n,f}^{(i)}$, are updated by decomposing $\hat{\mathscr{R}}_{n,f}^{(i)}$ using Eqs. (9.74) and (9.75).

Because the above EM Algorithm is performed independently in each frequency bin, a set of parameters estimated for a source at a frequency bin are not associated with those at different frequency bins in any sense. To perform BSS over all frequency bins, we need to combine the parameters estimated at different frequency bins into groups over all the frequency bins so that each group is composed of the parameters corresponding to a source. This operation is referred to as *permutation alignment* in the figure, and conducted after MLSE. There are many useful techniques for this purpose, such as that proposed in [22].

Finally, *MMSE filtering* is applied to the captured signals, $\mathbf{x}_{n,f}$, to obtain the estimated desired signal, $\hat{\mathbf{z}}_{n,f}^{(i)}$ for all i. This is identical to the filtering based on the multichannel Wiener filter performed in the above E-step by Eqs. (9.71) and (9.72).

9.5.2 MAPSE for BSS (MAP-BSS)

Now, we extend ML-BSS to MAP-BSS by introducing spectral priors modeled by LPS-GMMs as the generative models of individual sources. Because of the introduction of the spectral priors, MAP-BSS can estimate the power spectra of each desired signal more reliably and thus achieve more reliable BSS.

9.5.2.1 Generative Models for MAP-BSS

The generative models for MAP-BSS are almost identical to those for ML-BSS except that MAP-BSS includes the spectral priors for individual sources. As discussed in Sect. 9.4, each spectral prior is introduced to model the log-power spectra of each source, and the LPS-GMM is defined as in Eqs. (9.34), (9.35), (9.40), and (9.41). With the source index, i, the LPS-GMM is re-defined for each source as

$$p(\boldsymbol{\rho}_n^{(i)}) = \sum_{k=1}^{K} p(\boldsymbol{\rho}_n^{(i)}, k_n^{(i)} = k), \tag{9.76}$$

$$p(\boldsymbol{\rho}_n^{(i)}, k_n^{(i)} = k) = \lambda_k^{(i)} \prod_f p(\rho_{n,f}^{(i)} | k_n^{(i)} = k), \tag{9.77}$$

$$p(\rho_{n,f}^{(i)} | k_n^{(i)} = k) = \mathcal{N}^{(1)}(\rho_{n,f}^{(i)}; \mu_{k,f}^{(i)}, {\sigma_{k,f}^2}^{(i)}). \tag{9.78}$$

In this chapter, we assume all the model parameters of the GMMs, namely $\lambda_k^{(i)}$, $\mu_{k,f}^{(i)}$, and ${\sigma_{k,f}^2}^{(i)}$ for all i, k, and f are given in advance based on prior training using sound databases.

9.5.2.2 MAPSE Based on EM Algorithm

Let $\boldsymbol{\rho}$ and $\mathbf{R}$ be a set of log-power spectra $\rho^{(i)}$ and that of normalized spatial correlation matrices $\mathbf{R}^{(i)}$ for all the sources i, and $\Theta = \{\boldsymbol{\rho}, \mathbf{R}\}$ be the parameters to be estimated. Then, the MAP function for the MAP-BSS is defined as follows:

$$\mathscr{M}(\Theta) = p(\mathbf{x}|\boldsymbol{\rho}, \mathbf{R})p(\boldsymbol{\rho}). \tag{9.79}$$

MAPSE is achieved by obtaining Θ that maximizes the MAP function as

$$\hat{\Theta} = \arg\max_{\Theta} \mathscr{M}(\Theta). \tag{9.80}$$

As with MLSE, we do not have a closed form solution for the above maximization, and we adopt an optimization scheme that alternately updates $\Theta^{(i)}$ for each i by fixing $\Theta^{(i')}$ for $i' \neq i$ at their previously updated values, denoted by $\hat{\Theta}^{(i')}$, as in Eq. (9.81).

$$\hat{\Theta}^{(i)} = \arg\max_{\Theta^{(i)}} \mathscr{M}(\hat{\Theta}^{(1)}, \ldots, \hat{\Theta}^{(i-1)}, \Theta^{(i)}, \hat{\Theta}^{(i+1)}, \ldots, \hat{\Theta}^{(N_s)}). \tag{9.81}$$

Then, we can use the EM algorithm for the update in Eq. (9.81). We use the desired signal, $\mathbf{z}^{(i)}$, and the index of the Gaussian that is activated at each time frame n, namely $k_n^{(i)}$, as the hidden variables, and omit constant terms, and define and rewrite the Q-function for the EM algorithm as follows:

$$Q^{(i)}(\Theta^{(i)}|\hat{\Theta}) = E\{\log p(\mathbf{x}, \mathbf{z}^{(i)}, \boldsymbol{\rho}^{(i)}, k_n^{(i)}|\mathbf{R}^{(i)}, \hat{\theta}^{(i)})|\hat{\Theta}\}, \tag{9.82}$$

$$= \int_{-\infty}^{\infty} p(\mathbf{z}^{(i)}|\mathbf{x}, \hat{\boldsymbol{\rho}}^{(i)}, \hat{\mathbf{R}}^{(i)}, \hat{\theta}^{(i)}) \log p(\mathbf{x}, \mathbf{z}^{(i)}|\boldsymbol{\rho}^{(i)}, \mathbf{R}^{(i)}, \hat{\theta}^{(i)}) d\mathbf{z}^{(i)}$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)}) \log p(\boldsymbol{\rho}_n^{(i)}, k_n^{(i)} = k), \tag{9.83}$$

Similar to Eq. (9.70) and based on Eq. (9.77), we obtain

$$Q^{(i)}(\Theta^{(i)}|\hat{\Theta}) = -\sum_{n,f} \left\{ \frac{\mathrm{tr}\left(\left(\mathbf{R}_f^{(i)} \right)^{-1} \hat{\mathscr{R}}_{n,f}^{(i)} \right)}{\exp(\rho_{n,f}^{(i)})} + M\rho_{n,f}^{(i)} + \log\det \mathbf{R}_f^{(i)} \right\}$$

$$+ \sum_{n,f} \sum_{k} p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)}) \log p(\rho_{n,f}|k_n^{(i)} = k)$$

$$+ \sum_{n} \sum_{k} p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)}) \log \lambda_k^{(i)}, \tag{9.84}$$

where $\hat{\mathscr{R}}_{n,f}^{(i)} = E\{\mathbf{z}_{n,f}^{(i)}\mathbf{z}_{n,f}^{(i)\,H}|\mathbf{x}, \hat{\Theta}\}$ is the posterior expectation of the desired signal's spatial correlation matrix obtained by using the following equations.

$$\mathbf{W}_{n,f}^{(i)} = \exp(\hat{\rho}_{n,f}^{(i)})\hat{\mathbf{R}}_f^{(i)} \left(\hat{\mathbf{R}}_{n,f}^{(\mathbf{x})}\right)^{-1}, \tag{9.85}$$

$$\hat{\mathbf{z}}_{n,f}^{(i)} = \mathbf{W}_{n,f}^{(i)}\mathbf{x}_{n,f}, \tag{9.86}$$

$$\hat{\mathscr{R}}_{n,f}^{(i)} = \hat{\mathbf{z}}_{n,f}^{(i)}\hat{\mathbf{z}}_{n,f}^{(i)\,H} + (\mathbf{I} - \mathbf{W}_{n,f}^{(i)})\exp(\hat{\rho}_{n,f}^{(i)})\hat{\mathbf{R}}^{(i)}, \tag{9.87}$$

and $p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)})$ and $\log p(\rho_{n,f}|k_n^{(i)} = k)$ can further be rewritten as

$$p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)}) = \frac{\lambda_k^{(i)}\mathscr{N}^{(L)}(\hat{\boldsymbol{\rho}}_n^{(i)}; \boldsymbol{\mu}_k^{(i)}, \boldsymbol{\Sigma}_k^{(i)})}{\sum_{k'=1}^{K}\lambda_{k'}^{(i)}\mathscr{N}^{(L)}(\hat{\boldsymbol{\rho}}_n^{(i)}; \boldsymbol{\mu}_{k'}^{(i)}, \boldsymbol{\Sigma}_{k'}^{(i)})}, \tag{9.88}$$

$$\log p(\rho_{n,f}|k_n^{(i)} = k) = -\frac{(\rho_{n,f}^{(i)} - \mu_{k,f}^{(i)})^2}{2\sigma_{k,f}^{2\,(i)}} - \frac{1}{2}\log(\sigma_{k,f}^{2\,(i)}) - \frac{1}{2}\log 2\pi. \tag{9.89}$$

Here, $\mu_{k,f}^{(i)}$ and $\sigma_{k,f}^{2\,(i)}$ are the f-th element of $\boldsymbol{\mu}_k^{(i)}$ and the f-th diagonal component of $\boldsymbol{\Sigma}_k^{(i)}$ in Eq. (9.78), respectively.

Using the same form as in Eqs. (9.52) and (9.53), and omitting constant terms, the Q-function can further be rewritten as

$$Q(\Theta^{(i)}|\hat{\Theta}) = \sum_{f=0}^{F-1}\sum_{n=1}^{N} Q_{n,f}^{(i)}(\Theta_f^{(i)}|\hat{\Theta}) + \sum_{n=1}^{N}\sum_{k=1}^{K} p(k_n = k|\hat{\boldsymbol{\rho}}_n^{(i)})\log\lambda_k^{(i)}, \tag{9.90}$$

$$\begin{aligned} Q_{n,f}^{(i)}(\Theta_f^{(i)}|\hat{\Theta}) = &-\frac{\mathrm{tr}\left(\left(\mathbf{R}_f^{(i)}\right)^{-1}\hat{\mathscr{R}}_{n,f}^{(i)}\right)}{\exp(\rho_{n,f}^{(i)})} - M\rho_{n,f}^{(i)} - \log\det\mathbf{R}_f^{(i)} \\ &- \sum_{k=1}^{K} p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)})\frac{(\rho_{n,f}^{(i)} - \mu_{k,f}^{(i)})^2}{2\sigma_{k,f}^{2\,(i)}}. \end{aligned} \tag{9.91}$$

With the above Q-function, we can update $\Theta^{(i)}$ as in Eq. (9.81) by iterating the E-step and the M-step. In the E-step, we obtain the posterior expectation of the spatial correlation matrix, $\hat{\mathscr{R}}_{n,f}^{(i)} = E\{\mathbf{z}_{n,f}^{(i)}\mathbf{z}_{n,f}^{(i)\,H}|\mathbf{x}, \hat{\Theta}\}$ for all TF points, and the posterior distribution of k_n, $p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)})$, for all time frames n by using Eqs. (9.85)–(9.88). In the M-step, we update $\hat{\rho}_{n,f}^{(i)}$ and $\hat{\mathbf{R}}_f^{(i)}$ of the source i for all n and f as those that maximize Eq. (9.90). Assuming $\rho_{n,f}^{(i)} = \hat{\rho}_{n,f}^{(i)}$ to be fixed, $\hat{\mathbf{R}}_f^{(i)}$ can be updated in a way similar to that for MLSE as follows:

$$\hat{\mathbf{R}}_f^{(i)} = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{\exp(\hat{\rho}_{n,f}^{(i)})} \hat{\mathscr{R}}_{n,f}^{(i)}. \tag{9.92}$$

To update $\hat{\rho}_{n,f}^{(i)}$, we need to find $\rho_{n,f}^{(i)}$ that maximizes Eq. (9.91), which includes a nonlinear term, $\exp(\rho_{n,f}^{(i)})$. For this maximization, we use the non-linear optimization technique presented in Sect. 9.4.4. Assuming $\mathbf{R}_f^{(i)} = \hat{\mathbf{R}}_f^{(i)}$ to be fixed and setting the first derivative of Eq. (9.91) that should be zero as $\partial Q_{n,f}/\partial \rho_{n,f} = 0$, we can rewrite the resultant equation into the following form.

$$\exp(u) + u + \beta = 0, \tag{9.93}$$

where

$$u = -\rho_{n,f}^{(i)} + \alpha, \tag{9.94}$$

$$\beta = \alpha \left(-M + \sum_{k=1}^{K} \frac{p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)}) {\mu_{k,f}^{(i)}}^2}{{\sigma_{k,f}^2}^{(i)}} \right) \left(\sum_{k=1}^{K} \frac{p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)})}{{\sigma_{k,f}^2}^{(i)}} \right)^{-1} \tag{9.95}$$

$$\alpha = \log \mathrm{tr} \left(\left(\hat{\mathbf{R}}_f^{(i)} \right)^{-1} \hat{\mathscr{R}}_{n,f}^{(i)} \right) - \log \sum_{k=1}^{K} \frac{p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)})}{{\sigma_{k,f}^2}^{(i)}}. \tag{9.96}$$

The u value that satisfies Eq. (9.93) can be obtained by the method described in Sect. 9.4.4, and then the updated $\hat{\rho}_{n,f}^{(i)}$ can be obtained with the estimated value, $\hat{u}$, using Eq. (9.94).

9.5.2.3 Processing Flow of MAP-BSS Based on EM Algorithm

Figure 9.7 and Algorithm 4 show the processing flow of MAP-BSS based on the EM algorithm. The flow is slightly different from that of ML-BSS due to the introduction of the LPS-GMM, but the difference is not large. It is summarized as follows:

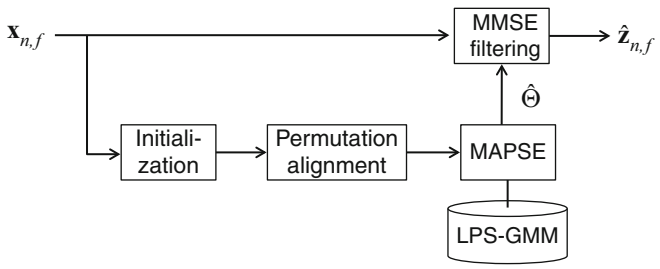

Fig. 9.7 Block diagram of MAP-BSS and MLMAP-BSS

Algorithm 4 Processing flow of MAP-BSS

1. Initialize $\hat{\boldsymbol{\rho}}$ and $\hat{\mathbf{R}}$, for example, based on the method proposed in [9]
2. Perform permutation alignment, e.g., based on the method proposed in [21].
3. Iterate the following steps until convergence is obtained.
   a. For each source i, perform the following steps.
      i. (E-step1) Obtain $\hat{\mathscr{R}}_{n,f}^{(i)}$ based on Eqs. (9.85)–(9.87) for each TF point.
      ii. (E-step2) Obtain $p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)})$ based on (9.88) for each time frame.
      iii. (M-step1) Update $\hat{\mathbf{R}}_{n,f}^{(i)}$ for each TF point based on Eq. (9.92).
      iv. (M-step2) For each TF point, obtain u that satisfies Eq. (9.93) based on the Newton–Raphson method (Sect. 9.4.4), and update $\hat{\rho}_{n,f}^{(i)}$ using Eq. (9.94).
4. Perform multichannel Wiener filtering based on Eqs. (9.85) and (9.86) for all i, n, and f, to separate the captured signal, $\mathbf{x}_{n,f}$, into individual desired signals, $\hat{\mathbf{z}}_{n,f}^{(i)}$.

- $\boldsymbol{\rho}$ is used as the parameters to be estimated instead of $\mathbf{v}$. Note, however, that most of the update equations for MAP-BSS are identical to those of ML-BSS based on the relationship $\rho_{n,f}^{(i)} = \log(v_{n,f}^{(i)})$.
- We need to perform the permutation alignment before the spectral estimation by MAPSE. This is because MAPSE uses the LPS-GMM that models speech spectra over all frequency bins as a whole based on Eq. (9.34). Instead, we do not need to perform the permutation alignment after MAPSE for MAP-BSS.
- In the E-step of MAP-BSS, we also calculate the posterior distribution of k_n given $\boldsymbol{\rho}_n = \hat{\boldsymbol{\rho}}_n$, namely $p(k_n^{(i)} = k|\hat{\boldsymbol{\rho}}_n^{(i)})$.
- In the M-step of MAP-BSS, $\hat{\rho}_{n,f}^{(1)}$ is updated based not only on the likelihood function but also on the spectral prior using the Newton–Raphson method.

9.5.2.4 Initialization of $\hat{\mathbf{R}}$ and $\hat{\boldsymbol{\rho}}$ (or $\hat{\mathbf{v}}$)

When we use an iterative optimization scheme such as the EM algorithm, it is desirable to start the iteration from the initial values for parameters that are close to the global maximum point. Such initialization may make the convergence faster, and may work favorably to avoid cases where the iterative optimization converges to local maximum points that are far from the global maximum point.

For this purpose, we adopt the following two initialization schemes in this chapter.

1. It was reported that the use of direction feature clustering proposed in [22] is useful for the initialization of $\hat{\mathbf{R}}$ and $\hat{\mathbf{v}}$ for ML-BSS [9]. So, we adopt this scheme not only for the initialization of ML-BSS but also for the initialization of $\hat{\mathbf{R}}$ and $\hat{\boldsymbol{\rho}}$ for MAP-BSS.

2. In addition, as shown in the experimental section, the use of $\hat{\mathbf{R}}$ and $\hat{\mathbf{v}}$ estimated by ML-BSS for the initialization of MAP-BSS can further improve the accuracy of the source separation by MAP-BSS.

Hereafter, we refer to MAPSE based BSS with the above initialization scheme 1 simply as MAP-BSS, and that with the above initialization scheme 2 as MLMAP-BSS. We compare the performance of the two schemes in the next section.

9.6 Experiments

This section describes two experiments that we conducted to examine the effectiveness of ML-BSS, MAP-BSS, and MLMAP-BSS.

The first experiment compares the performance of the three methods under various recording conditions. We used three different reverberation conditions and two different source number conditions, including an underdetermined condition, in this experiment.

The second experiment evaluates the three methods using a standard evaluation task for BSS, taken from the 2010 Signal Separation Evaluation Campaign, SiSEC-2010 [28].

9.6.1 Evaluation 1 with Aurora-2 Speech Database

We first evaluated the performance of the three methods using various sound mixtures that are generated by mixing continuous digit utterances randomly extracted from Aurora-2 database [18] under different reverberation conditions. Table 9.1 summarizes the conditions used for the experiments. The number of Gaussians for LPS-GMM, namely 256, was determined based on our preliminary experiments so that we can obtain a sufficiently accurate speech model with relatively low computational cost. The number of EM iterations, namely 100, was selected because the EM algorithm converged in most cases after 100 iterations in our preliminary experiments. Note that the number was set equally for all three methods. This means that, for MLMAP-BSS, the total number of EM iterations for initialization by ML-BSS and EM iterations by MAP-BSS was set at 100. Figure 9.8 shows the condition that we set for measuring the impulse responses used to generate sound mixtures for the test. We used audio speakers at 70° and 150° for the sound mixtures composed of two sources, and used audio speakers at 70°, 150°, and 245° for the sound mixtures composed of three sources.

Table 9.1 Experimental conditions for Evaluation 1

Sampling frequency	8 kHz
# of microphones	2
# of sources	2, 3
Microphone spacing	4 cm
# of test mixtures composed of two sources	30
# of test mixtures composed of three sources	40
Average length of test mixture	5 s
Reverberation time (T_{60})	110, 220, 400 ms
Analysis/synthesis window	Hanning
Window length	128 ms (1,024 points)
Window shift	32 ms (256 points)
# of EM iterations	100
Training data for LPS-GMM	Clean training data in Aurora-2
Size of training data for LPS-GMM	8,440 utterances
# of Gaussians used for LPS-GMM	256

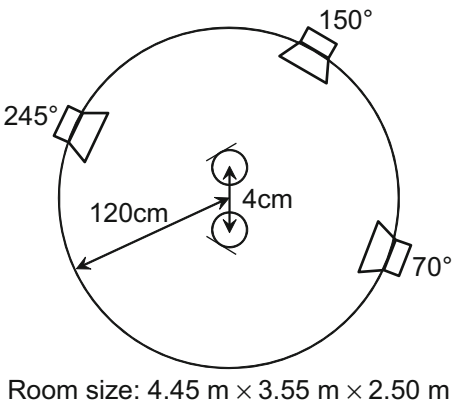

Fig. 9.8 Setting for measuring impulse responses

We used the following four measures, SDR, ISR, SIR, and SAR, proposed in [28] to evaluate the BSS performance.

Name of measure	Type of distortion to be evaluated
Signal to Distortion Ratio (SDR)	Total distortion composed of the following three types of distortion
Source Image to Spatial distortion Ratio (ISR)	Linear distortion
Source to Interference Ratio (SIR)	Remaining interference
Sources to Artifacts Ratio (SAR)	Nonlinear distortion

For all these measures, higher values mean better performance.

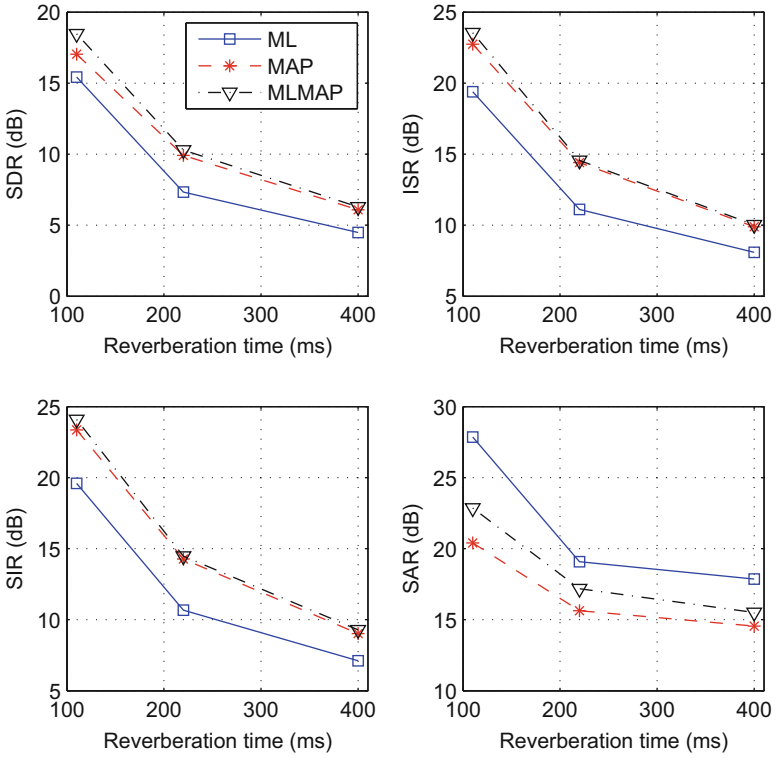

Fig. 9.9 SDR, ISR, SIR, and SAR obtained for Evaluation 1 using ML-BSS (ML), MAP-BSS (MAP), MLMAP-BSS (MLMAP) on mixtures composed of two sources. Each value is averaged over all the utterances separated by BSS

Figure 9.9 shows the evaluation results obtained under the three different reverberation conditions when we used sound mixtures obtained from two sources for the test. The figure shows that the two proposed methods, MAP-BSS and MLMAP-BSS, both substantially outperformed ML-BSS in terms of all the measures except for SAR. This suggests that the use of the LPS-GMM improved the separation performance. In addition, when we listen to the separated sounds, audible artifacts and nonlinear distortion are less prominent with MAP-BSS and MLMAP-BSS than with ML-BSS, although SAR is lower for MAP-BSS and MLMAP-BSS than ML-BSS. This characteristic of the two proposed methods may be explained as follows: While the MAP-BSS and MLMAP-BSS nonlinearly modified the spectral shapes of the desired signals by the nonlinear optimization realized using LPS-GMMs, MAP-BSS and MLMAP-BSS can still improve the audible quality of the separated signals thanks to the use of the spectral priors modeled by LPS-GMMs.

Figure 9.10 shows the evaluation results when we used sound mixtures obtained from three sources for the test. The figure confirms that MLMAP-BSS again substantially outperformed ML-BSS, but MAP-BSS did not outperform ML-BSS.

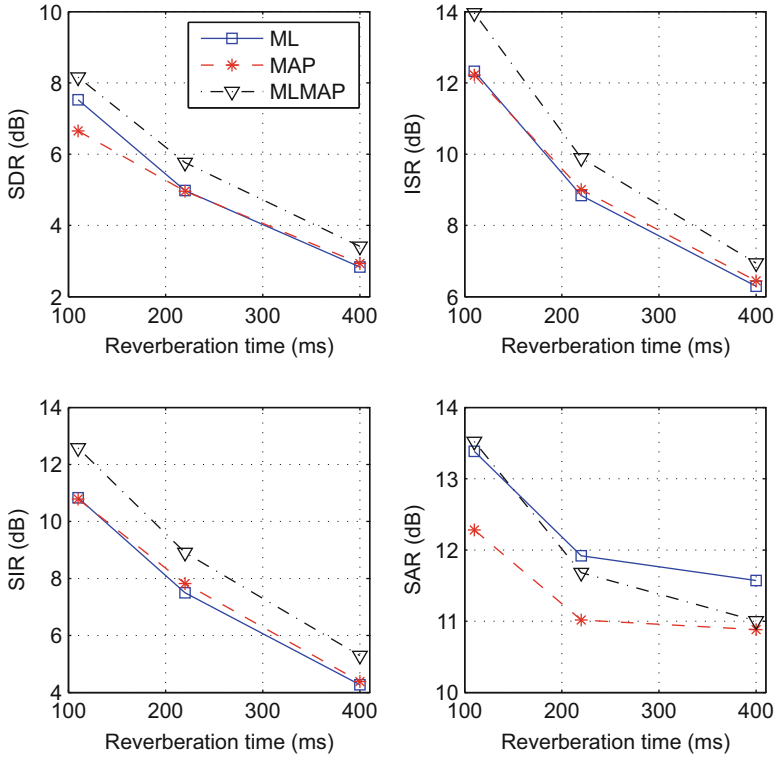

Fig. 9.10 SDR, ISR, SIR, and SAR obtained for Evaluation 1 using ML-BSS (ML), MAP-BSS (MAP), MLMAP-BSS (MLMAP) on mixtures composed of three sources. Each value is averaged over all the utterances separated by BSS

The difference between MAP-BSS and MLMAP-BSS is in the initialization methods, and thus this result suggests that the performance of MAPSE based BSS depends largely on how it is initialized. In spite of this characteristic, we always obtained a substantial performance improvement with MLMAP-BSS by comparison with one of the state-of-the-art BSS technique, ML-BSS.

9.6.2 Evaluation 2 with SiSEC Database

Next, we evaluated the performance of ML-BSS, MAP-BSS, and MLMAP-BSS, using a standard data set for BSS evaluation, namely the live recordings in the development set1 extracted from the underdetermined speech mixtures in the SiSEC-2010 evaluation task. Table 9.2 summarizes in more details the conditions of the evaluation using the data set.

Table 9.2 Experimental condition for Evaluation 2

Sampling frequency	16 kHz
# of microphones	2
# of sources	3
Microphone spacing	5 cm
# of test mixtures composed of three sources	2 (one composed of three male speakers and the other composed of three female speakers)
Average length of test mixture	10 s
Reverberation time (T_{60})	250 ms
Analysis/synthesis window	Hanning
Window length	128 ms (2,048 points)
Window shift	32 ms (512 points)
# of EM iterations	100
Training data for LPS-GMM	A clean speech data set composed of Japanese and multilingual utterances
Size of training data for LPS-GMM	1,000 utterances
# of Gaussians used for LPS-GMM	256

The training data for LPS-GMM were extracted from Aurora-2

Table 9.3 SDR, ISR, SIR, and SAR obtained for Evaluation 2 on mixtures composed of three sources

		Male			Female		
		Speaker			Speaker		
		1	2	3	1	2	3
ML-BSS	SDR (dB)	2.0	1.1	5.6	5.3	4.6	5.9
	ISR (dB)	5.6	3.5	11.0	6.9	10.0	11.4
	SIR (dB)	2.5	0.0	7.3	10.8	5.7	7.7
	SAR (dB)	8.3	6.3	10.4	10.6	9.3	11.2
MAP-BSS	SDR (dB)	**2.8**	**1.4**	4.5	4.8	3.2	4.5
	ISR (dB)	**8.0**	3.2	10.0	6.6	8.7	8.8
	SIR (dB)	**3.8**	**2.6**	6.5	9.9	4.4	6.6
	SAR (dB)	**9.2**	5.1	9.6	10.2	8.9	10.1
MLMAP-BSS	SDR (dB)	**3.4**	**2.0**	4.8	**5.5**	**4.6**	**6.1**
	ISR (dB)	**9.7**	**4.0**	8.7	**7.3**	**10.3**	**11.8**
	SIR (dB)	**4.2**	**3.5**	**8.0**	**11.3**	**6.3**	**8.3**
	SAR (dB)	**10.0**	5.3	8.8	**10.7**	9.2	11.0

Bold fonts in the table highlights scores of MAP-BSS or those of MLMAP-BSS that outnumbered those of ML-BSS

The evaluation results, namely SDR, ISR, SIR, and SAR obtained by ML-BSS, MAP-BSS, and MLMAP-BSS using sound mixtures composed of three sources are shown in Table 9.3. By comparing the proposed approaches (MAP-BSS and MLMAP-BSS) with the conventional approach (ML-BSS), we confirmed that MLMAP-BSS again substantially outperformed ML-BSS in most cases in terms

of all the measures except for SAR, but MAP-BSS did not outperform ML-BSS in many cases. This result is almost the same as that obtained for Evaluation 1 using sound mixtures composed of three sources. This again indicates that the introduction of the spectral priors can improve the performance of ML-BSS, but the performance of MAPSE based BSS depends largely on how it is initialized.

In summary, MLMAP-BSS can be a good approach because it can substantially outperform ML-BSS in most cases by effectively mitigating the sensitivity of MAPSE based BSS to the initialization.

9.7 Concluding Remarks

This chapter described a versatile technique for extending a widely used multichannel speech enhancement approach, referred to as maximum-likelihood spectral estimation (MLSE) based speech enhancement, to maximum a posteriori spectral estimation (MAPSE) based speech enhancement. While MLSE mainly uses the spatial features of the signals for speech enhancement, MAPSE also uses the speech spectral features and priors to achieve more reliable and accurate spectral estimation. This chapter also proposed a method that uses Gaussian mixture models for speech log-power spectra (LPS-GMMs) as useful spectral priors for MAPSE. Because an LPS-GMM can accurately model the complex distribution of the speech spectra, the use of LPS-GMMs can improve the accuracy of MAPSE. As a concrete application of MAPSE, this chapter described a method for extending MLSE based BSS (ML-BSS) to MAPSE based BSS (MAP-BSS/MLMAP-BSS). Although ML-BSS is known as a state-of-the-art underdetermined BSS technique, MAP-BSS/MLMAP-BSS can achieve more accurate BSS based on the use of spectral priors. Our experiments showed that MLMAP-BSS can stably and substantially outperform ML-BSS under various recording conditions, including underdetermined conditions, in most cases.

References

1. J. Benesty, S. Makino, J. Chen (eds.), *Speech Enhancement (Signals and Communication Technology)* (Springer, Berlin, 2005)
2. C.M. Biship, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer, New York, 2010)
3. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. Ser. B Methodol. **39**, 1–38 (1977)
4. N.Q.K. Duong, E. Vincent, R. Gribonval, Under-determined reverberant audio source separation using a full-rank spatial covariance model. IEEE Trans. Audio Speech Lang. Process. **18**(7), 1830–1840 (2010)
5. M. Fujimoto, T. Nakatani, Model-based noise suppression using unsupervised estimation of hidden Markov model for non-stationary noise, in *Proceedings of INTERSPEECH 2013* (2013), pp. 2982–2986

6. S. Gannot, M. Moonen, Subspace methods for multimicrophone speech dereverberation. EURASIP J. Adv. Signal Process. **2003**(11), 1074–1090 (2003)
7. J.F. Gemmeke, T. Virtanen, A. Hurmalainen, Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Trans. Audio Speech Lang. Process. **19**(7), 2067–2080 (2011)
8. S. Haykin, *Adaptive Filter Theory*, 5th edn. (Prentice Hall, Englewood Cliffs, 2013)
9. K. Iso, S. Araki, S. Makino, T. Nakatani, H. Sawada, T. Yamada, A. Nakamura, Blind source separation of mixed speech in a high reverberation environment, in *Proceedings of 3rd Joint Workshop on Hands-free Speech Communication and Microphone Array (HSCMA-2011)* (2011), pp. 36–39
10. N. Ito, S. Araki, T. Nakatani, Probabilistic integration of diffuse noise suppression and dereverberation, in *Proceedings of IEEE ICASSP-2014* (2014), pp. 5204–5208
11. Y. Iwata, T. Nakatani, Introduction of speech log-spectral priors into dereverberation based on Itakura-Saito distance minimization, in *Proceedings of IEEE ICASSP-2012* (2012), pp. 245–248
12. Y. Iwata, T. Nakatani, M. Fujimoto, T. Yoshioka, H. Saito, MAP spectral estimation of speech using log-spectral prior for noise reduction (in Japanese), in *Proceedings of Autumn-2012 Meeting of the Acoustical Society of Japan* (2012), pp. 795–798
13. Y. Izumi, N. Ono, S. Sagayama, Sparseness-based 2ch BSS using the EM algorithm in reverberant environment, in *Proceedings of IEEE WASPAA-2007* (2007), pp. 147–150
14. P.C. Loizou, *Speech Enhancement: Theory and Practice*, 2nd edn. (CRC Press, Boca Raton, 2013)
15. P.J. Moreno, B. Raj, R.M. Stern, A vector taylor series approach for environment-independent speech recognition, in *Proceedings of IEEE ICASSP-1996*, vol. 2 (1996), pp. 733–736
16. T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, B.H. Juang, Speech dereverberation based on variance-normalized delayed linear prediction. IEEE Trans. Audio Speech Lang. Process. **18**(7), 1717–1731 (2010)
17. A. Ogawa, K. Kinoshita, T. Hori, T. Nakatani, A. Nakamura, Fast segment search for corpus-based speech enhancement based on speech recognition technology, in *Proceedings of IEEE ICASSP-2014* (2014), pp. 1576–1580
18. D. Pearce, H.G. Hirsch, The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions, in *Proceedings of INTERSPEECH-2000*, vol. 2000 (2000), pp. 29–32
19. M. Rainer, Noise power spectral density estimation based on optimal smoothing and minimum statistics. IEEE Trans. Speech Audio Process. **9**(5), 504–512 (2001)
20. S.J. Rennie, J.R. Hershey, P.A. Olsen, Single-channel multitalker speech recognition. IEEE SP Mag. **27**(6), 66–80 (2010)
21. H. Sawada, S. Araki, R. Mukai, S. Makino, Grouping separated frequency components by estimating propagation model parameters in frequency-domain blind source separation. IEEE Trans. Audio Speech Lang. Process. **15**(5), 1592–1604 (2007)
22. H. Sawada, S. Araki, S. Makino, Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment. IEEE Trans. Audio Speech Lang. Process. **19**(3), 516–527 (2011)
23. M. Seltzer, D. Yu, Y. Wang, An investigation of deep neural networks for noise robust speech recognition, in *Proceedings of IEEE ICASSP-2013* (2013), pp. 7398–7402
24. M. Souden, J. Chen, J. Benesty, S. Affes, An integrated solution for online multichannel noise tracking and reduction. IEEE Trans. Audio Speech Lang. Process. **19**, 2159–2169 (2011)
25. M. Togami, Y. Kawaguchi, R. Takeda, Y. Obuchi, N. Nukaga, Optimized speech dereverberation from probabilistic perspective for time varying acoustic transfer function. IEEE Trans. Audio Speech Lang. Process. **21**(7), 1369–1380 (2013)

26. O. Yilmaz, S. Rickard, Blind separation of speech mixture via time-frequency masking. IEEE Trans. Signal Process. **52**(7), 1830–1847 (2004)
27. T. Yoshioka, T. Nakatani, M. Miyoshi, H.G. Okuno, Blind separation and dereverberation of speech mixtures by joint optimization. IEEE Trans. Audio Speech Lang. Process. **19**(1), 69–84 (2011)
28. E. Vincent, H. Sawada, P. Bofill, S. Makino, J. Rosca, First stereo audio source separation evaluation campaign: data, algorithms and results, in *Proceedings of International Conference on Independent Component Analysis (ICA)*, pp. 552–559 (2007)

Chapter 10
Modulation Processing for Speech Enhancement

Kuldip Paliwal and Belinda Schwerin

Abstract Many of the traditionally speech enhancement methods reduce noise from corrupted speech by processing the magnitude spectrum in a short-time Fourier analysis-modification-synthesis (AMS) based framework. More recently, use of the modulation domain for speech processing has been investigated, however early efforts in this direction did not account for the changing properties of the modulation spectrum across time. Motivated by this and evidence of the significance of the modulation domain, we investigated the processing of the modulation spectrum on a short-time basis for speech enhancement. For this purpose, a modulation domain-based AMS framework was used, in which the trajectories of each acoustic frequency bin were processed frame-wise in a secondary AMS framework. A number of different enhancement algorithms were investigated for the enhancement of speech in the short-time modulation domain. These included spectral subtraction and MMSE magnitude estimation. In each case, the respective algorithm was used to modify the short-time modulation magnitude spectrum within the modulation AMS framework. Here we review the findings of this investigation, comparing the quality of stimuli enhanced using these modulation based approaches to stimuli enhanced using corresponding modification algorithms applied in the acoustic domain. Results presented show modulation domain based approaches to have improved quality compared to their acoustic domain counterparts. Further, MMSE modulation magnitude estimation (MME) is shown to have improved speech quality compared to Modulation spectral subtraction (ModSSub) stimuli. MME stimuli are found to have good removal of noise without the introduction of musical noise, problematic in spectral subtraction based enhancement. Results also show that ModSSub has minimal musical noise compared to acoustic Spectral subtraction, for appropriately selected modulation frame duration. For modulation domain based methods, modulation frame duration is shown to be an important parameter, with quality generally improved by use of shorter frame durations. From the results of experiments conducted, it is concluded that the short-time modulation domain provides an effective alternative to the short-time acoustic domain for speech

K. Paliwal (✉) • B. Schwerin
Griffith School of Engineering, Nathan Campus, Griffith University, Brisbane, QLD, Australia
e-mail: k.paliwal@griffith.edu.au

T. Ogunfunmi et al. (eds.), *Speech and Audio Processing for Coding, Enhancement and Recognition*, DOI 10.1007/978-1-4939-1456-2_10

processing. Further, that in this domain, MME provides effective noise suppression without the introduction of musical noise distortion.

10.1 Introduction

Speech enhancement aims to improve the quality of noisy speech, typically by suppressing noise in such a way that the residual noise is not annoying to listeners, and speech distortion introduced by the enhanced process is minimised. There is an extensive range of methods for speech enhancement in the literature, many of which can be broadly classified as being spectral subtraction, statistical (MMSE), Kalman filtering, Wiener filtering, or subspace based methods. The first two of these are particularly well known, in part for being simple yet effective for enhancing speech corrupted with additive noise distortion.

Spectral subtraction [7, 8, 33] is perhaps one of the earliest and most extensively studied speech enhancement methods for the removal of additive noise. While particularly effective at suppressing background noise, it does, however, result in the introduction of perceptually annoying spectral artefacts referred to as musical noise.

Overcoming this problem is the MMSE (minimum mean-square error) short-time spectral amplitude estimator (referred to here as the acoustic magnitude estimator (AME)) of Ephraim and Malah [14]. The good performance of AME has been largely attributed to the use of the decision-directed approach for estimation of the a priori signal-to-noise ratio (SNR) [9, 51]. Despite not being quite as effective as spectral subtraction at suppressing noise, the colourless nature of its residual distortion has resulted in the AME method remaining one of the most effective and popular methods for speech enhancement in the acoustic domain.

Many of the popular single-channel speech enhancement methods in the literature, including the above mentioned methods, perform enhancement in the acoustic spectral domain within a short-time Fourier (acoustic) analysis-modification-synthesis (AMS) framework. More recently, however, the modulation domain has gained popularity for speech processing.

This popularity has been in part due to psychoacoustic and physiological evidence supporting the significance of the modulation domain for the analysis of speech signals. For example, the experiments of Bacon and Grantham [6] showed that there are channels in the auditory system which are tuned for the detection of modulation frequencies. Sheft and Yost [56] showed that our perception of temporal dynamics corresponds to our perceptual filtering into modulation frequency channels and that faithful representation of these modulations is critical to our perception of speech. Experiments of Schreiner and Urbas [52] showed that a neural representation of amplitude modulation is preserved through all levels of the mammalian auditory system, including the highest level of audition, the auditory cortex. Neurons in the auditory cortex are thought to decompose the acoustic

spectrum into spectro-temporal modulation content [39], and are best driven by sounds that combine both spectral and temporal modulations [11, 32, 54].

Further, low frequency modulations of sound have been shown to be the fundamental carriers of information in speech [4]. Drullman et al. [12, 13], for example, investigated the importance of modulation frequencies for intelligibility by applying low-pass and high-pass filters to the temporal envelopes of acoustic frequency subbands. They showed frequencies between 4 and 16 Hz to be important for intelligibility, with the region around 4–5 Hz being the most significant. In a similar study, Arai et al. [2] showed that applying band-pass filters between 1 and 16 Hz does not impair speech intelligibility. While the envelope of the acoustic magnitude spectrum represents the shape of the vocal tract, the modulation spectrum represents how the vocal tract changes as a function of time. It is these temporal changes that convey most of the linguistic information (or intelligibility) of speech. In the above intelligibility studies, the lower limit of 1 Hz stems from the fact that the slow vocal tract changes do not convey much linguistic information. In addition, the lower limit helps to make speech communication more robust, since the majority of noises occurring in nature vary slowly as a function of time and hence their modulation spectrum is dominated by modulation frequencies below 1 Hz. The upper limit of 16 Hz is due to the physiological limitation on how fast the vocal tract is able to change with time.

At this point it is useful to differentiate the acoustic spectrum from the modulation spectrum as follows. The acoustic spectrum is the short-time Fourier transform (STFT) of the speech signal, while the modulation spectrum at a given acoustic frequency is the STFT of the time series of the acoustic spectral magnitudes at that frequency. The short-time modulation spectrum is thus a function of time, acoustic frequency and modulation frequency.

Modulation domain processing has found applications in areas such as speech coding [3, 5, 60], speech recognition [21, 26, 29, 35, 40, 61, 66], speaker recognition [30, 31, 37, 64], and objective speech intelligibility evaluation [17, 19, 27, 45, 59]. While short-time processing in the modulation domain has been used for automatic speech recognition (ASR) [27, 29, 61], early efforts to utilise the modulation domain for speech enhancement has assumed speech and noise to be stationary, applying fixed filtering on the trajectories of the acoustic magnitude spectrum. For example, Hermansky et al. [22] proposed band-pass filtering the time trajectories of the cubic-root compressed short-time power spectrum to enhance speech. Falk et al. [16] and Lyons and Paliwal [36] applied similar band-pass filtering to the time trajectories of the short-time magnitude (power) spectrum for speech enhancement.

There are two main limitations associated with typical modulation filtering methods. First, they use a filter design based on the long-term properties of the speech modulation spectrum, while ignoring the properties of noise. As a consequence, they fail to eliminate noise components present within the speech modulation regions. Second, the modulation filter is fixed and applied to the entire signal, even though the properties of speech and noise change over time. To capture this nonstationarity, speech can instead be assumed quasi-stationary, and the trajectories of the acoustic

magnitude spectrum can be processed on a short-time basis. Therefore, through use of short-time modulation domain processing, these limitations can be addressed.

Assuming noise to be additive, we have therefore investigated different speech enhancement approaches within a framework which facilitates modification of the short-time modulation spectrum. Methods evaluated include spectral subtraction, MMSE magnitude estimation, Wiener and Kalman filtering in the modulation domain—however details of Wiener and Kalman filtering methods are not included here but can be found in [42] and [57]. Details of the modulation AMS framework, and modulation spectral subtraction and MMSE modulation magnitude estimation enhancement methods are reviewed in the following section. This is followed by a review of findings from experiments comparing the quality of stimuli processed using various acoustic and modulation domain enhancement methods.

10.2 Methods

10.2.1 Modulation AMS-Based Framework

As mentioned in the introduction, many frequency domain based speech enhancement methods are based on the (acoustic) short-time Fourier AMS framework [1,7, 10, 14]. The traditional acoustic AMS procedure for speech enhancement includes three stages: the analysis stage (where noisy speech is processed using STFT analysis); the modification stage (where the noisy spectrum is modified to compensate for noise and distortion); and the synthesis stage (where an inverse STFT operation, followed by overlap-add synthesis is used to reconstruct the enhanced signal).

The modulation-domain based enhancement methods described here, instead make use of a modulation AMS-based framework, in which the traditional acoustic AMS-based framework is extended to the modulation domain, facilitating modification of the short-time modulation magnitude spectrum to improve speech quality. In this framework each frequency component of the acoustic magnitude spectra, obtained during the analysis stage of the acoustic AMS procedure, is processed frame-wise across time using a secondary AMS procedure in the modulation domain. This dual AMS framework we denote the modulation AMS framework, and is described as follows.

Let us assume an additive noise model in which clean speech is corrupted by uncorrelated additive noise to produce noisy speech as given by

$$x(n) = s(n) + d(n), \tag{10.1}$$

where $x(n)$, $s(n)$, and $d(n)$ are the noisy speech, clean speech, and noise signals, respectively, and n denotes a discrete-time index. Although speech is non-stationary, it can be assumed quasi-stationary, and therefore the noisy speech signal can be processed frame-wise using the running STFT analysis [62] given by

$$X(l,k) = \sum_{n=0}^{N-1} x(n + lZ)\, w(n)\, e^{-j2\pi nk/N}, \tag{10.2}$$

where l refers to the acoustic frame index, k refers to the index of the acoustic frequency, N is the acoustic frame duration (AFD) in samples, Z is the acoustic frame shift (AFS) in samples, and $w(n)$ is the acoustic analysis window function. Speech processing typically uses the Hamming analysis window, and an AFD of 20–40 ms and an AFS of 10–20 ms [24, 34, 43, 46, 49].

Using STFT analysis, we can represent Eq. (10.1) by

$$X(l,k) = S(l,k) + D(l,k), \tag{10.3}$$

where $X(l,k)$, $S(l,k)$, and $D(l,k)$ are the STFTs of the noisy speech, clean speech and noise signals, respectively. Each of these can be expressed in terms of their acoustic magnitude spectrum and acoustic phase spectrum. In polar form, the STFT of the noisy speech signal, for example, can be expressed as

$$X(l,k) = |X(l,k)|\, e^{\,j\angle X(l,k)}, \tag{10.4}$$

where $|X(l,k)|$ denotes the acoustic magnitude spectrum,[1] $\angle X(l,k)$ denotes the acoustic phase spectrum, and the discrete-time signal $x(n)$ is completely characterised by its magnitude and phase spectra.

Traditional AMS-based speech enhancement methods modify or enhance the noisy acoustic magnitude spectrum $|X(l,k)|$, while keeping the noisy acoustic phase spectrum unchanged. One reason for this is that, for Hamming-windowed frames (of 20–40 ms duration), the phase spectrum is considered relatively unimportant for speech enhancement [55, 65]. Similarly, in the modulation AMS framework, the noisy acoustic phase spectra is left unchanged, and the noisy acoustic magnitude spectrum is modified by processing the time trajectories of each frequency component of the acoustic magnitude spectra frame-wise in a second AMS procedure as follows.

Note that traditionally, the modulation spectrum has been computed as the Fourier transform of the intensity envelope of the band-pass filtered signal [12, 17, 23]. However here, we utilise the short-time Fourier transform (STFT) instead of band-pass filtering. In the acoustic STFT domain, the quantity closest to the intensity envelope of a band-pass filtered signal is the magnitude-squared spectrum. However, more recent works [16, 28] support the suitability of using the time trajectories of the short-time acoustic magnitude spectrum for computation of

[1] Note that for references made to the magnitude, phase or complex spectra throughout this text, the STFT modifier is implied unless otherwise stated. The acoustic and modulation modifiers are also included to disambiguate between acoustic and modulation domains.

the short-time modulation spectrum. Therefore either the acoustic magnitude or magnitude-squared spectra can be used for computation of the modulation spectrum.

Thus, the running STFT is used to compute the modulation spectrum from the acoustic magnitude spectrum as

$$\mathcal{X}(\ell, k, m) = \sum_{l=0}^{\mathcal{N}-1} \left| X_{l+\ell\mathcal{Z}}(k) \right| v(l)\, e^{-j\,2\pi lm/\mathcal{N}}, \tag{10.5}$$

where ℓ is the modulation frame index, k is the index of the acoustic frequency, m refers to the index of the modulation frequency, $\mathcal{N}$ is the modulation frame duration (MFD) in terms of acoustic frames, $\mathcal{Z}$ is the modulation frame shift (MFS) in terms of acoustic frames, and $v(l)$ is the modulation analysis window function. The modulation spectrum can be written in polar form as

$$\mathcal{X}(\ell, k, m) = |\mathcal{X}(\ell, k, m)|\, e^{\,j\angle\mathcal{X}(\ell,k,m)}, \tag{10.6}$$

where $|\mathcal{X}(\ell, k, m)|$ is the modulation magnitude spectrum, and $\angle\mathcal{X}(\ell, k, m)$ is the modulation phase spectrum.

For methods presented here, the modulation magnitude spectrum of clean speech is estimated from the noisy modulation magnitude spectrum, while the noisy modulation phase spectrum $\angle\mathcal{X}(\ell, k, m)$ is left unchanged. The modified modulation spectrum is then given by

$$\mathcal{Y}(\ell, k, m) = \left|\hat{\mathcal{S}}(\ell, k, m)\right| e^{\,j\angle\mathcal{X}(\ell,k,m)}, \tag{10.7}$$

where $\left|\hat{\mathcal{S}}(\ell, k, m)\right|$ is an estimate of the clean modulation magnitude spectrum. Equation (10.7) can also be written in terms of spectral gain function $\mathcal{G}(\ell, k, m)$ applied to the modulation spectrum of noisy speech as follows

$$\mathcal{Y}(\ell, k, m) = \mathcal{G}(\ell, k, m)\, \mathcal{X}(\ell, k, m), \tag{10.8}$$

where

$$\mathcal{G}(\ell, k, m) = \frac{\left|\hat{\mathcal{S}}(\ell, k, m)\right|}{|\mathcal{X}(\ell, k, m)|}. \tag{10.9}$$

The inverse STFT operation, followed by least-squares overlap-add synthesis [48], are then used to compute the modified acoustic magnitude spectrum as given by

$$|Y(l, k)| = \sum_{\ell} \left\{ v_s(l - \ell\mathcal{Z}) \sum_{m=0}^{\mathcal{N}-1} \mathcal{Y}(\ell, k, m)\, e^{\,j\,2\pi(l-\ell\mathcal{Z})m/\mathcal{N}} \right\}, \tag{10.10}$$

where $v_s(\ell)$ is a (modulation) synthesis window function.

The modified acoustic magnitude spectrum is combined with the noisy acoustic phase spectrum, to produce the modified acoustic spectrum as follows

$$Y(l,k) = |Y(l,k)|\, e^{j\angle X(l,k)}. \tag{10.11}$$

The enhanced speech signal is constructed by applying the inverse STFT operation, followed by least-squares overlap-add synthesis, to the modified acoustic spectrum giving

$$y(n) = \sum_{l} \left\{ w_s\,(n - lZ) \sum_{k=0}^{N-1} Y(l,k)\, e^{\,j2\pi(n-lZ)k/N} \right\}, \tag{10.12}$$

where $w_s(l)$ is the (acoustic) synthesis window function. The modified Hanning window [20] was used for both the acoustic and modulation synthesis windows. A block diagram of the AMS-based framework for speech enhancement in the short-time spectral modulation domain is shown in Fig. 10.1.

10.2.2 Modulation Spectral Subtraction

Classical spectral subtraction is an intuitive and effective speech enhancement method utilising a short-time Fourier AMS framework, and enhancing speech by subtracting a spectral estimate of noise from the noisy speech spectrum in either the magnitude or energy domain.

The modulation spectral subtraction method (ModSSub) [44] similarly utilises the above modulation AMS framework. Within the modification stage of this framework, the noisy modulation magnitude spectrum $\left|\mathcal{X}(\ell,k,m)\right|$ is replaced with an estimate of the clean modulation magnitude spectrum, calculated using a spectral subtraction rule similar to the one proposed by Berouti et al. [7], and given by Eq. (10.13).

$$\left|\hat{\mathcal{S}}(\ell,k,m)\right| = \begin{cases} \left(\Delta(\ell,k,m)\right)^{\frac{1}{\gamma}}, & \text{if } \Delta(\ell,k,m) \geq \beta\left|\hat{\mathcal{D}}(\ell,k,m)\right|^{\gamma} \\ \left(\beta\left|\hat{\mathcal{D}}(\ell,k,m)\right|^{\gamma}\right)^{\frac{1}{\gamma}}, & \text{otherwise,} \end{cases} \tag{10.13}$$

where

$$\Delta(\ell,k,m) = \left|\mathcal{X}(\ell,k,m)\right|^{\gamma} - \rho\left|\hat{\mathcal{D}}(\ell,k,m)\right|^{\gamma}. \tag{10.14}$$

Here, β is the spectral floor parameter used to set spectral magnitude values falling below the spectral floor $\left(\beta\left|\hat{\mathcal{D}}(\ell,k,m)\right|^{\gamma}\right)^{\frac{1}{\gamma}}$, to that spectral floor; and γ determines the subtraction domain (e.g., for γ set to unity the subtraction is performed in the

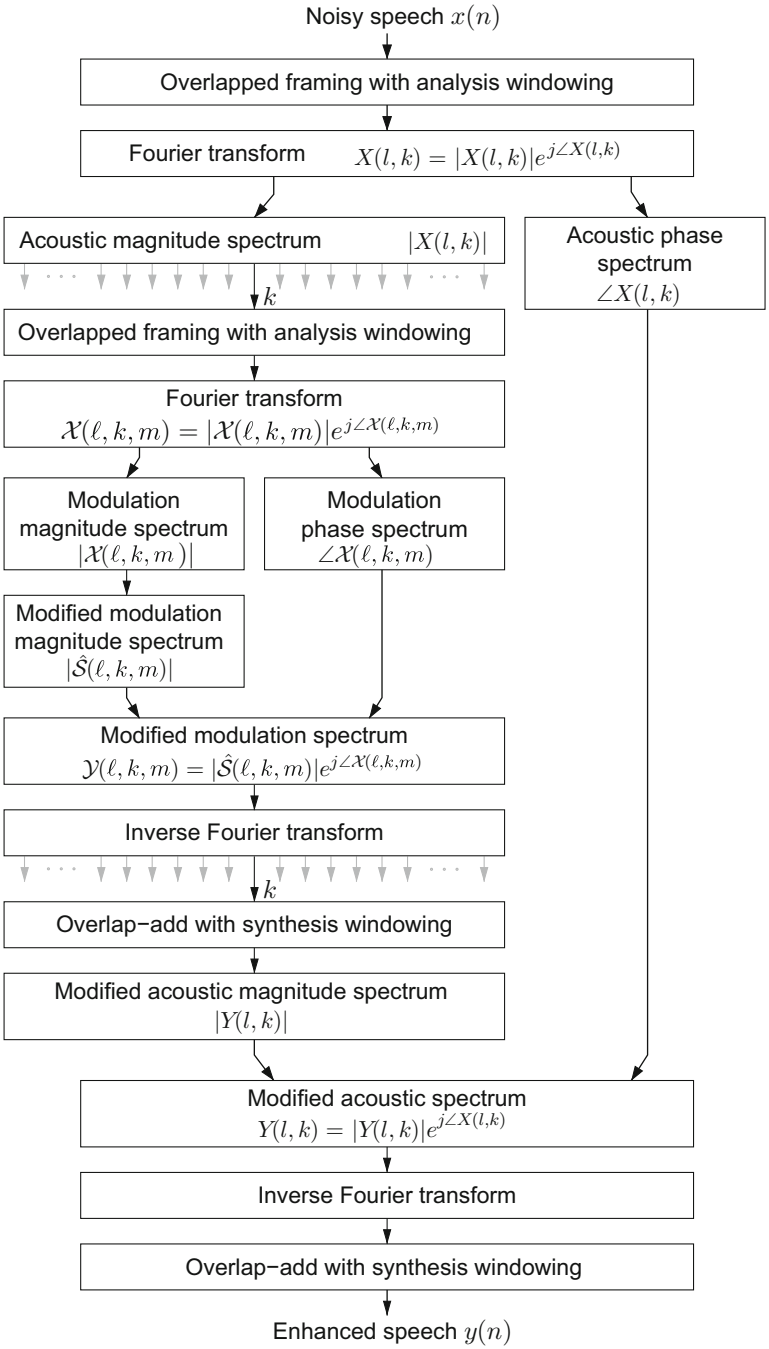

Fig. 10.1 Block diagram of the modulation AMS framework for speech enhancement in the short-time spectral modulation domain

magnitude spectral domain, while for $\gamma = 2$ the subtraction is performed in the magnitude-squared spectral domain). The subtraction factor, ρ, governs the amount of over-subtraction applied, and is calculated as (following Berouti et al. [7])

$$\rho = \begin{cases} 5, & SNR < -5 \\ \rho_0 - \frac{SNR}{s}, & -5 \leq SNR \leq 20 \\ 1, & SNR > 20 \end{cases} \tag{10.15}$$

where ρ_0 is the value for ρ at an SNR of 0 dB, and $\frac{1}{s}$ is the slope of the line from ρ_0 at 0 dB to $\rho = 1$ at 20 dB.

The estimate of the modulation magnitude spectrum of the noise, denoted by $\left|\hat{\mathcal{D}}(\ell, k, m)\right|$, is obtained based on a decision from a simple voice activity detector (VAD) [34], applied in the modulation domain. The VAD classifies each modulation domain segment as either 1 (*speech present*) or 0 (*speech absent*), using the following binary rule

$$\Phi(\eta, k) = \begin{cases} 1, & \text{if } \phi(\eta, k) \geq \theta \\ 0, & \text{otherwise} \end{cases}, \tag{10.16}$$

where θ is an empirically determined speech presence threshold, and $\phi(\eta, k)$ denotes a modulation segmental signal-to-noise ratio (SNR) computed as follows:

$$\phi(\eta, k) = 10 \log_{10} \left(\frac{\sum_m \left|\mathcal{X}(\ell, k, m)\right|^2}{\sum_m \left|\hat{\mathcal{D}}(\ell - 1, k, m)\right|^2} \right). \tag{10.17}$$

The noise estimate is updated during speech absence using the following averaging rule [63]

$$\left|\hat{\mathcal{D}}(\ell, k, m)\right|^\gamma = \lambda \left|\hat{\mathcal{D}}(\ell - 1, k, m)\right|^\gamma + (1 - \lambda) \left|\mathcal{X}(\ell, k, m)\right|^\gamma, \tag{10.18}$$

where λ is a forgetting factor chosen according to the stationarity of the noise.

Parameters of ModSSub were determined empirically via listening tests with stimuli constructed using a range of parameter values (see [44] for further details). Those values found to work best are given in Table 10.1.

Table 10.1 Parameter values applied to ModSSub

Parameter	Value (ms)	Parameter	Value
AFD	32	γ	2
AFS	8	θ	3 dB
MFD	256	λ	0.98
MFS	32	β	0.002
		ρ_0	4

In the ModSSub method, the frame duration used for computing the short-time modulation spectrum (MFD) was found to be an important parameter, providing a trade-off between musical noise distortion and spectral smearing distortion types. Durations of 220–256 ms were found to provide the best trade-off, minimising the musical noise apparent but with some introduction of audible spectral smearing distortion. The disadvantages of using longer modulation domain analysis window are as follows. Firstly, we are assuming stationarity which we know is not the case. Secondly, quite a long portion is needed for the initial estimation of noise, and thirdly, as shown by [41], speech quality and intelligibility is higher when the modulation magnitude spectrum is processed using short frame durations and lower when processed using longer frame durations. These findings suggested that use of MMSE magnitude estimation approach, which does not introduce musical noise distortion (for appropriately selected smoothing parameter) and therefore may be applied using shorter MFD, would be better suited to processing in the short-time modulation domain.

10.2.3 MMSE Modulation Magnitude Estimation

The minimum mean-square error short-time spectral amplitude estimator of Ephraim and Malah [14] has been employed in the past for speech enhancement in the acoustic frequency domain with much success, offering the advantage of effective noise suppression without the introduction of perceptually annoying musical noise. Consequently, MMSE magnitude estimation was similarly investigated in the short-time modulation domain, and found to be particular effective. Again, the modulation AMS-based framework was utilised, and suppression of noise from the modulation magnitude spectrum was performed as follows.

The MMSE modulation magnitude estimator (MME) [42] estimates the modulation magnitude spectrum of clean speech from noisy observations, minimising the mean-square error between the modulation magnitude spectra of clean and estimated speech

$$\epsilon = \mathrm{E}\left[\left(|\mathcal{S}(\ell,k,m)| - |\hat{\mathcal{S}}(\ell,k,m)|\right)^2\right] \tag{10.19}$$

where $\mathrm{E}[\,\cdot\,]$ denotes the expectation operator. Closed form solution to this problem in the acoustic spectral domain has been reported by Ephraim and Malah [14] under the assumptions that speech and noise are additive in the time domain, and that their individual short-time spectral components are statistically independent, identically distributed, zero-mean Gaussian random variables.

To apply MMSE processing in the modulation domain, we need to make similar assumptions, namely that (1) speech and noise are additive in the short-time acoustic spectral magnitude domain, i.e.,

$$|X(l,k)| = |S(l,k)| + |D(l,k)| \tag{10.20}$$

and (2) the individual short-time modulation spectral components of $\mathcal{S}(\ell,k,m)$ and $\mathcal{D}(\ell,k,m)$ are independent, identically distributed Gaussian random variables. The reasoning for the first assumption is that at high SNRs (greater than around 8 dB) the phase spectrum remains largely unchanged by additive noise distortion [34, p. 104]. For the second assumption, we can apply an argument similar to that of Ephraim and Malah [14], where the central limit theorem is used to justify the statistical independence of spectral components of the Fourier transform. For the STFT, this assumption is valid only in the asymptotic sense, that is, when the frame duration is large. However, Ephraim and Malah have used an AFD of 32 ms in their formulation to achieve good results. For MMSE magnitude estimation in the modulation domain, we should also make the MFD to be as large as possible, however it must not be so large as to be adversely affected by the nonstationarity of the magnitude spectral sequence.

With the above assumptions in mind, the modulation magnitude spectrum of clean speech can be estimated from the noisy modulation spectrum under the MMSE criterion [following 14] as

$$\left|\hat{\mathcal{S}}(\ell,k,m)\right| = \mathrm{E}\Big[\,\big|\mathcal{S}(\ell,k,m)\big|\,\Big|\,\mathcal{X}(\ell,k,m)\,\Big] \tag{10.21}$$

$$= \mathcal{G}(\ell,k,m)\big|\mathcal{X}(\ell,k,m)\big| \tag{10.22}$$

where $\mathcal{G}(\ell,k,m)$ is the MME spectral gain function given by

$$\mathcal{G}(\ell,k,m) = \frac{\sqrt{\pi}}{2}\frac{\sqrt{\nu(\ell,k,m)}}{\gamma(\ell,k,m)}\,\Lambda\Big[\nu(\ell,k,m)\Big]. \tag{10.23}$$

Here, $\nu(\ell,k,m)$ defined as

$$\nu(\ell,k,m) \triangleq \frac{\xi(\ell,k,m)}{1+\xi(\ell,k,m)}\,\gamma(\ell,k,m) \tag{10.24}$$

and $\Lambda\big[\,\cdot\,\big]$ is the function

$$\Lambda\big[\theta\big] = \exp\left(-\frac{\theta}{2}\right)\left[(1+\theta)I_0\left(\frac{\theta}{2}\right) + \theta\, I_1\left(\frac{\theta}{2}\right)\right], \tag{10.25}$$

where $I_0\,(\cdot)$ and $I_1\,(\cdot)$ denote the modified Bessel functions of zero and first order, respectively. In the above equations $\xi(\ell,k,m)$ and $\gamma(\ell,k,m)$ are interpreted [after 38] as the a priori SNR, and the a posteriori SNR. These quantities are defined as

$$\xi(\ell,k,m) \triangleq \frac{\mathrm{E}\left[|\mathcal{S}(\ell,k,m)|^2\right]}{\mathrm{E}\left[|\mathcal{D}(\ell,k,m)|^2\right]} \tag{10.26}$$

and

$$\gamma(\ell,k,m) \triangleq \frac{|\mathcal{X}(\ell,k,m)|^2}{\mathrm{E}\left[|\mathcal{D}(\ell,k,m)|^2\right]}. \tag{10.27}$$

respectively.

Since in practice only noisy speech is observable, the $\xi(\ell,k,m)$ and $\gamma(\ell,k,m)$ parameters have to be estimated. For this we apply the decision-directed approach [14] in the short-time spectral modulation domain. In the decision-directed method the a priori SNR is estimated by recursive averaging as follows

$$\hat{\xi}(\ell,k,m) = \alpha \frac{\left|\hat{\mathcal{S}}(\ell-1,k,m)\right|^2}{\hat{\lambda}(\ell-1,k,m)} + (1-\alpha)\ \max\left[\hat{\gamma}(\ell,k,m)-1,0\right] \tag{10.28}$$

where α controls the trade-off between noise reduction and transient distortion [9, 14], $\hat{\lambda}(\ell,k,m)$ is an estimate of $\lambda(\ell,k,m) \triangleq \mathrm{E}\left[\left|\mathcal{D}(\ell,k,m)\right|^2\right]$, and the a posteriori SNR estimate is obtained by

$$\hat{\gamma}(\ell,k,m) = \frac{|\mathcal{X}(\ell,k,m)|^2}{\hat{\lambda}(\ell,k,m)}. \tag{10.29}$$

Note that limiting the minimum value of the a priori SNR has a considerable effect on the nature of the residual noise [9, 14], providing a trade-off between musical and white noise distortions. For this reason, a lower bound ξ_{min} is typically used to prevent a priori SNR estimates falling below a prescribed value, i.e.,

$$\hat{\xi}(\ell,k,m) = \max\left[\hat{\xi}(\ell,k,m), \xi_{min}\right]. \tag{10.30}$$

Modulation noise power spectral estimates are needed. For this, a simple procedure is employed, where an initial estimate of modulation power spectrum of noise is computed from six leading silence frames. This estimate is then updated during speech absence using a recursive averaging rule [51, 63], applied in the modulation spectral domain as follows

$$\hat{\lambda}(\ell,k,m) = \varphi\hat{\lambda}(\ell-1,k,m) + (1-\varphi)\,|\mathcal{X}(\ell,k,m)|^2 \tag{10.31}$$

where φ is a forgetting factor chosen depending on the stationarity of the noise. The speech presence or absence is determined using a statistical model-based voice

activity detection (VAD) algorithm (the decision-directed decision rule without hang-over) by Sohn et al. [58], applied in the modulation spectral domain.

In Ephraim and Malah's classical paper on acoustic magnitude estimation (AME), authors also proposed an AME formulation under the uncertainty of speech presence [14]. Here, the quality of enhanced speech was shown to be further improved (compared to that generated by AME alone), without introducing any additional distortions. In a later paper, they went on to show that applying AME to the log-magnitude spectrum [15], which is more suited to speech processing [18], also results in improved enhanced speech quality. Motivated by these observations, we also investigated the effect of applying speech presence uncertainty (SPU) and log-magnitude spectral processing to the MME formulation.

10.2.3.1 MMSE Modulation Magnitude Estimation with SPU

Using SPU, the optimal estimate of the modulation magnitude spectrum is given by the relation

$$\left|\hat{\mathcal{S}}(\ell,k,m)\right| = \phi(\ell,k,m)\mathcal{G}(\ell,k,m)\left|\mathcal{X}(\ell,k,m)\right|, \tag{10.32}$$

where $\mathcal{G}(\ell,k,m)$ is the MME spectral gain function given by Eq. (10.23), and $\phi(\ell,k,m)$ is given by

$$\phi(\ell,k,m) = \frac{\Lambda(\ell,k,m)}{1+\Lambda(\ell,k,m)}, \tag{10.33}$$

with

$$\Lambda(\ell,k,m) = \frac{(1-q_m)}{q_m}.\frac{\exp\left(\nu(\ell,k,m)\right)}{1+\hat{\xi}(\ell,k,m)}, \tag{10.34}$$

and $\nu(\ell,k,m)$ given by Eq. (10.24). Here q_m is the probability of signal presence in the mth spectral component, and is a tunable parameter. Applying $\left|\hat{\mathcal{S}}(\ell,k,m)\right|$ of Eq. (10.32) in the modulation AMS framework produced stimuli denoted type MME+SPU [42].

10.2.3.2 MMSE Log-Modulation Magnitude Estimation

Minimising the mean-squared error of the log-modulation magnitude spectrum, the optimal estimate of the modulation magnitude spectrum is given by the relation

$$\left|\hat{\mathcal{S}}(\ell,k,m)\right| = \mathscr{G}(\ell,k,m)\left|\mathcal{X}(\ell,k,m)\right|, \tag{10.35}$$

Table 10.2 Parameter values applied to MMSE modulation magnitude estimation based methods

	MME	MME+SPU	LogMME
AFD	32 ms	32 ms	32 ms
AFS	1 ms	1 ms	1 ms
MFD	32 ms	32 ms	32 ms
MFS	2 ms	2 ms	2 ms
q_m	–	0.3	–
$\xi_{\min}$	−25 dB	−25 dB	−25 dB
Smoothing parameter (α)	0.998	0.995	0.996

where $\mathcal{G}(\ell, k, m)$ is the spectral gain function given by

$$\mathcal{G}(\ell, k, m) = \nu(\ell, k, m)\exp\Big(\frac{1}{2}\mathrm{Ei}\big[\nu(\ell, k, m)\big]\Big), \tag{10.36}$$

$\mathrm{Ei}\big[\cdot\big]$ is the exponential integral, and $\nu(\ell, k, m)$ (a function of a priori and a posteriori SNRs) is given by Eq. (10.24). Stimuli of type LogMME [42] are then constructed by applying $\big|\hat{\mathcal{S}}(\ell, k, m)\big|$ given by Eq. (10.35) in the modulation AMS framework.

10.2.3.3 MME Parameters

Key parameters of MME, MME+SPU and LogMME were each determined subjectively via listening tests (see [42] for further details). Parameters found to work best are shown in Table 10.2.

10.3 Speech Quality Assessment

Enhancement methods such as those discussed in this chapter aim to improve the quality of speech degraded by additive noise distortion. To evaluate the effectiveness of different methods in achieving this objective, enhanced stimuli are typically evaluated by human listeners in subjective listing tests. A number of methodologies for conducting these tests can be used, and are generally classified as either ratings based, or preference based. For a detailed review of subjective testing methods, the interested reader is referred to [34, Chap. 10]. Evaluations of quality discussed in this chapter make use of AB listening tests to determine the preference of listeners. These tests play stimuli pairs, in randomised order, and listeners are asked to select their preference. Pair-wise scoring is then used to calculate a preference score for each treatment type (a detailed description of the procedure can be found in [42,44]).

However, subjective experiments are heavily dependent on the reliability and judgements of each listener, and are somewhat time consuming. Therefore objective

metrics, which compare enhanced and clean stimuli via some measure, are popular as a quick indicator of enhanced stimuli quality. There is an extensive number of metrics for this purpose available, each giving an indication of some aspects of speech quality, while neglecting other aspects. Loizou [34] and Quackenbush et al. [47] may be referenced for a detailed review of many of these. Two popular measures used to evaluate quality in the literature, and used here, are the PESQ metric [50] and the Segmental SNR [47].

10.4 Evaluation of Short-Time Modulation-Domain Based Methods with Respect to Quality

As previously mentioned, spectral subtraction (SpecSub) [8] and the MMSE acoustic magnitude estimator (AME) [14] are well known acoustic domain methods for the enhancement of stimuli corrupted with additive noise distortion. SpecSub is a particularly well known method, featuring effective suppression of background noise but having the disadvantage of introducing audibly distracting musical noise artefact to reconstructed stimuli. Addressing the problem of musical noise is the AME method, which does not introduce musical noise distortion (for appropriately selected a priori SNR estimation smoothing parameter). While the noise suppression of AME is not as effective as SpecSub, the absence of musical noise in reconstructed stimuli makes it preferred over SpecSub for the enhancement of speech.

Initial evaluation of the effectiveness of short-time modulation-domain processing therefore compared the quality of stimuli processed using AME, with that processed using modulation spectral subtraction (ModSSub) and MMSE modulation-magnitude estimation (MME) methods. Results of the subjective comparison (in the form of AB human listening tests) of the quality of processed stimuli are shown in Fig. 10.2. From these results we see that both ModSSub and MME improve on AME, suggesting that processing in the short-time modulation domain results in improved processed stimuli quality.

Noisy stimuli processed using ModSSub is noted to have improved noise suppression, as featured by SpecSub, but without significant introduction of musical noise. This improved noise suppression makes it preferred over AME by listeners. Since the effect of noise on speech is dependent on the frequency, and the SNR of noisy speech varies across the acoustic spectrum [25], it is reasonable to expect that the ModSSub method will also attain better performance for coloured noises than the acoustic spectral subtraction and AME methods. This is because one of the strengths of the method is that each subband is processed independently and thus it is the time trajectories in each subband that are important, and not the relative levels in-between bands at a given time instant. It is also for this reason that the modulation spectral subtraction method avoids much of the musical noise problem associated with acoustic spectral subtraction. However, as we have previously noted, the ModSSub method requires longer frame durations for modulation domain

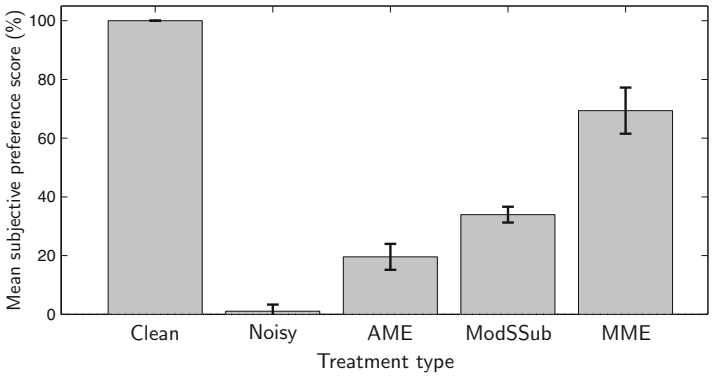

Fig. 10.2 Speech enhancement results for the subjective experiment comparing the quality of enhanced stimuli. The results are in terms of mean subjective preference scores (%) with standard error bars for (**a**) clean; (**b**) noisy (degraded at 5 dB Additive white Gaussian noise (AWGN)); and stimuli generated using the following treatment types: (**c**) AME [14]; (**d**) ModSSub [44]; and (**e**) MME [42]

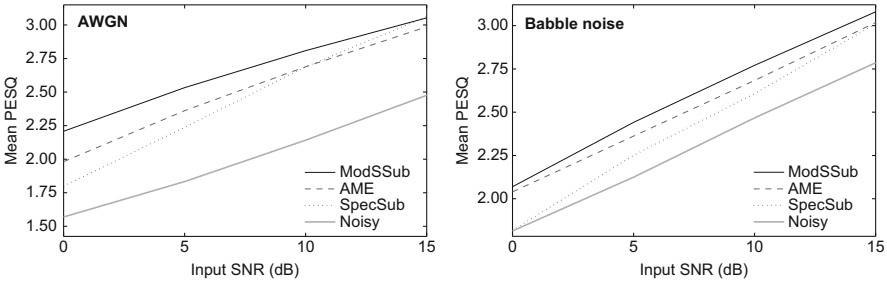

Fig. 10.3 Results of an objective quality experiment for AWGN (*left*) and babble (*right*) noise types. The results are in terms of mean PESQ scores as a function of input SNR (dB). Mean scores are shown for (**a**) noisy; and stimuli generated using the following treatment types: (**b**) SpecSub [8]; (**c**) AME [14]; and (**d**) ModSSub [44]

processing. This also means that longer non-speech durations are required to update noise estimates, and may result in the method being less adaptive to rapidly changing noise conditions. A comparison of performance of ModSSub for Additive white Gaussian noise (AWGN) and babble noise types, in terms of mean Perceptual evaluation of speech quality (PESQ, [50]) scores, are shown in Fig. 10.3. ModSSub scores higher than AME and SpecSub for both noise types. The lower scores of SpecSub, particularly at low input SNRs, again indicates that use of the modulation domain for spectral subtraction provides a considerable improvement for both stationary and non-stationary noise types.

However, as previously mentioned, in the ModSSub method, the MFD was found to be an important parameter, providing a trade-off between musical noise distortion and spectral smearing distortion types. MME, on the other hand, utilises MMSE magnitude estimation in the short-time modulation domain, an approach which

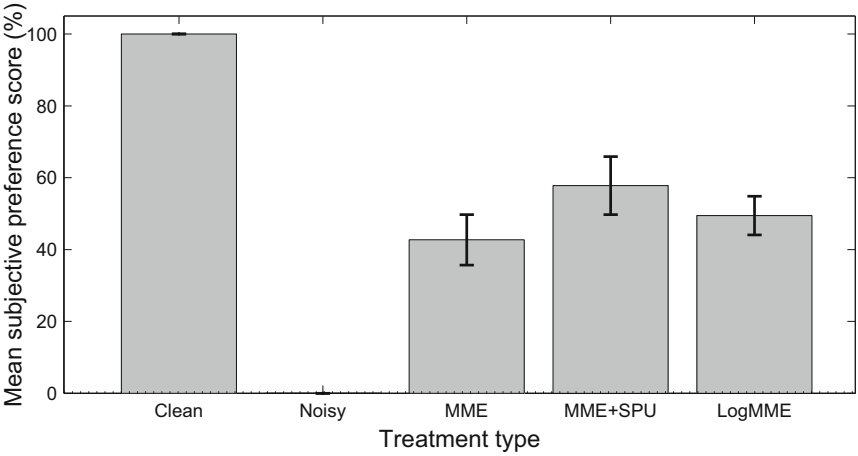

Fig. 10.4 Mean subjective preference scores (%) with standard error bars for (**a**) clean; (**b**) noisy (degraded at 5 dB Additive white Gaussian noise (AWGN)); and stimuli generated using the following treatment types: (**c**) MME [42]; (**d**) MME+SPU [42]; and (**e**) LogMME [42]

is not susceptible to musical noise in the same way that spectral subtraction is. Therefore much shorter MFDs could be used, overcoming the problem of spectral smearing. The result was a considerable quality improvement, as indicated by the much higher preference for MME stimuli in Fig. 10.2. As observed in the acoustic domain, while the noise suppression for MME was not quite as effective as for ModSSub, the greatly improved residual noise made it preferred by listeners. Comparing the sound of ModSSub and MME stimuli, ModSSub stimuli have less audible background noise, but there is spectral smearing heard as a type of slurring, and some low level musical-type noise. MME does not have these musical noise artefacts, while having improved removal of background noise compared to AME.

The important parameter of MME, like for AME, is the smoothing parameter for the decision-directed a priori SNR estimation. A relatively high smoothing parameter is required for MME, resulting in reducing adaptability to rapidly changing noise characteristics, and some smoothing, heard as a loss of crispness in speech.

Just as in the acoustic domain, MMSE magnitude estimation was improved with the use of speech presence uncertainly, or log-domain processing, so too was there improvement in the modulation domain. This can be seen in the results of subjective experiments shown in Fig. 10.4 comparing the quality of MME, to that of MME with speech presence uncertainty (MME+SPU) and modulation MMSE log-magnitude estimation (LogMME). Here we see that MME+SPU was generally preferred by listeners over MME and LogMME stimuli types. It is noted that the difference between the MME, LogMME and MME+SPU stimuli types is relatively small compared to that observed in the acoustic domain, and mainly heard as an improvement in background noise attenuation. It was also observed that the

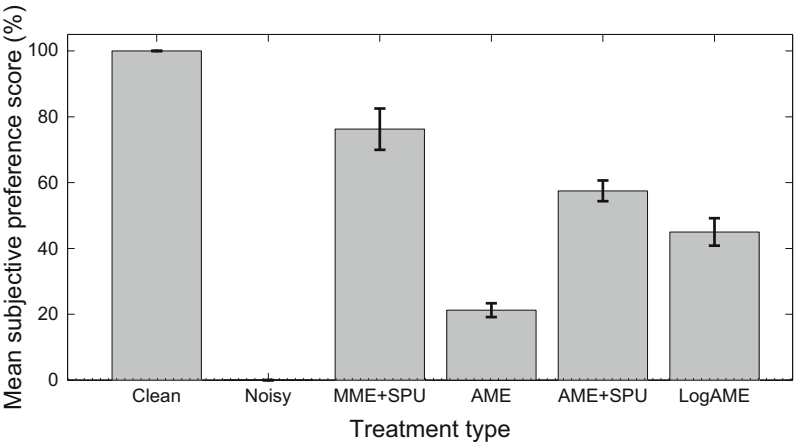

Fig. 10.5 Mean subjective preference scores (%) with standard error bars for (**a**) clean; (**b**) noisy (degraded at 5 dB Additive white Gaussian noise (AWGN)); and stimuli generated using the following treatment types: (**c**) MME+SPU [42]; (**d**) AME [14]; (**e**) AME+SPU [14]; and (**f**) LogAME [15]

improvement in speech quality due to the use of SPU was more noticeable in some stimuli than in others. For less-stationary noise types, MME+SPU and LogMME were found to be quite similar in quality, again offering a small improvement over MME.

To complete the comparison to acoustic domain MMSE magnitude estimation, a subjective comparison of stimuli quality of AME variations (including AME [14], AME+SPU [14], and LogAME [15]) to the MME+SPU method is shown in Fig. 10.5. As expected, AME+SPU was the most preferred of the acoustic AME-based methods, though scores for LogAME also indicate improved quality compared to AME. Comparing with MME+SPU, MME+SPU provided improved removal of noise compared to all AME-based methods, resulting in higher preference by listeners. Otherwise, the nature of residual noise is quite similar and colourless.

As mentioned in the introduction, speech enhancement methods can generally be classified as either spectral subtraction, statistical (MMSE), Wiener filtering, Kalman filtering, or subspace based methods. In this chapter, we have evaluated the performance of both spectral subtraction and MMSE magnitude estimation in the modulation domain. To complete an evaluation of these methods, we compare the quality of stimuli processed using these methods, with that enhanced using Wiener filtering and Kalman filtering in the modulation domain. Wiener filtering has recently been investigated in the short-time modulation domain (ModWiener) utilizing a non-iterative Wiener filtering approach based on the a priori approach of Scalart and Filho [51]. Modulation-domain Kalman filtering (ModKalman) has also been investigated in So and Paliwal [57]. There, a coloured-noise Kalman

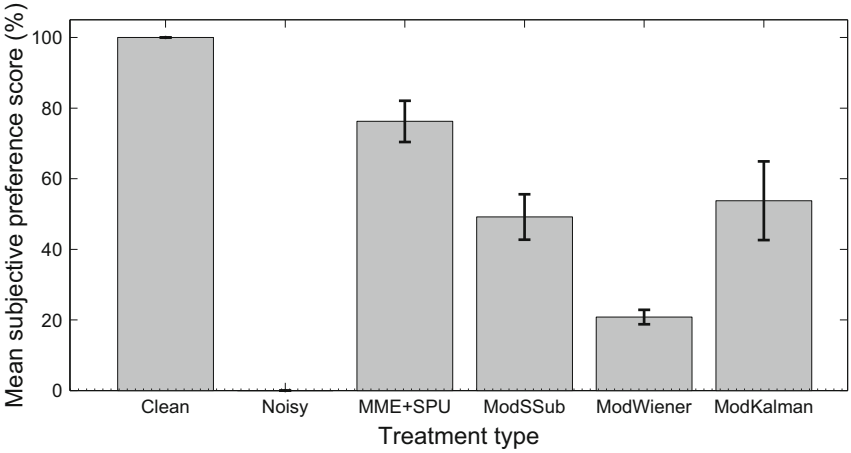

Fig. 10.6 Mean subjective preference scores (%) including standard error bars for (**a**) clean; (**b**) noisy (degraded at 5 dB AWGN); and stimuli generated using the following treatment types: (**c**) MME+SPU [42]; (**d**) ModSSub [44]; (**e**) ModWiener [42]; and (**f**) ModKalman [57]

filtering approach was applied to each temporal trajectory, with AME enhanced speech used to estimate the initial LPCs.

The resulting mean preference scores for the subjective comparison and AWGN are shown in Fig. 10.6. Of the modulation-domain based approaches, MME+SPU was clearly preferred by listeners over other treatment types, and ModKalman was the next most preferred. It is noted that some listeners recorded a similar preference for ModSSub and ModKalman, while others preferred ModKalman over ModSSub. The good performance of the MME+SPU and ModKalman methods is partly attributed to their use of small MFDs, which is consistent with the findings reported in [41]. ModWiener was the least preferred of the investigated enhancement methods.

Subjective experiments for stimuli corrupted with a range of coloured noise types were also conducted. Results of a subjective experiment utilising babble noise is shown in Fig. 10.7, and show consistent results to those observed for AWGN.

The preferences shown in Figs. 10.6 and 10.7 are well explained by looking at the spectrograms of each stimuli type. Spectrograms of the utterance, "The sky that morning was clear and bright blue", by a male speaker are shown in Fig. 10.8. For type MME+SPU stimuli (shown in Fig. 10.8c), we can see there is good background noise removal, less residual noise, no musical-type noise, and no visible spectral smearing. ModKalman stimuli (Fig. 10.8f) also have good background noise removal and no visible spectral smearing, but has clear dark spots throughout the background heard as a musical type noise. ModSSub stimuli (Fig. 10.8d), on the other hand, have less musical type noise than ModKalman but also contains spectral smearing due to the use of longer frame durations, causing distortion

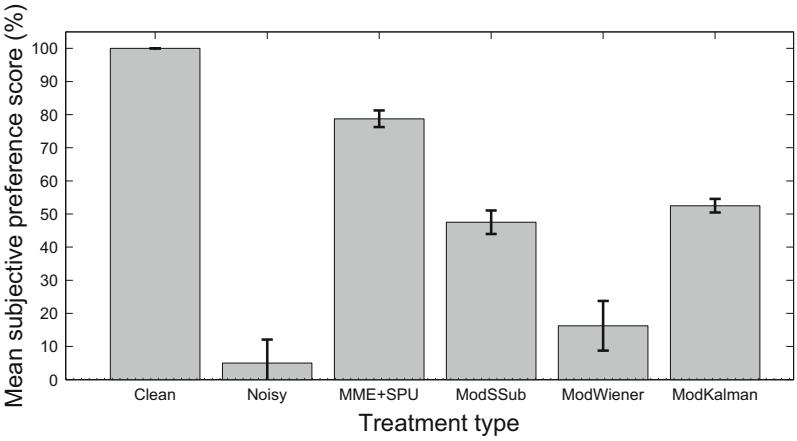

Fig. 10.7 Mean subjective preference scores (%) for (**a**) clean; (**b**) noisy (degraded with babble noise at 5 dB); and stimuli generated using the following treatment types: (**c**) MME+SPU [42]; (**d**) ModSSub [44]; (**e**) ModWiener [42]; and (**f**) ModKalman [57]

in the processed speech. ModWiener, which was the least effective method, had considerable distortion in the stimuli, seen as darkness in the background of its spectrogram (Fig. 10.8e). The poor performance of ModWiener was in part due to difficulty tuning, where parameters working better for one stimuli was considerably different for another.

Objective evaluations of the quality of stimuli enhanced using each of the modulation domain methods were also conducted. Here, all 30 stimuli from the Noizeus corpus, corrupted with the indicated noise type at each input SNR level, were enhanced using each of the modulation domain based enhancement methods. Quality for each stimuli, compared to clean, were evaluated using segmental SNR and PESQ measures. Mean scores were calculated for each treatment type, noise type, and input SNR. Figure 10.9 shows mean segmental SNRs for (a) AWGN and (b) babble noise types. Similarly, Fig. 10.10 shows mean PESQ scores.

The segmental SNRs are demonstrated to have the highest correlation with subjective results, with MME+SPU generally scoring higher than other methods. While for stimuli corrupted with white noise, objective scores for MME+SPU, ModSSub and ModKalman were quite close, there was a bigger difference in scores when considering coloured noise stimuli. Here, MME+SPU scored somewhat higher than other methods. For babble noise, ModSSub and ModKalman were very close with ModSSub scoring a little higher, but it is noted that for some other noise types such as street noise, ModKalman scored higher than ModSSub. Overall, segmental SNRs are consistent with the findings of the coloured noise subjective experiments.

PESQ scores are found to show less consistency with subjective results. Here, MME+SPU scores higher at high SNRs, but ModSSub scores higher at lower SNRs (including 5 dB, the input SNR used for subjective tests). Results indicate that the

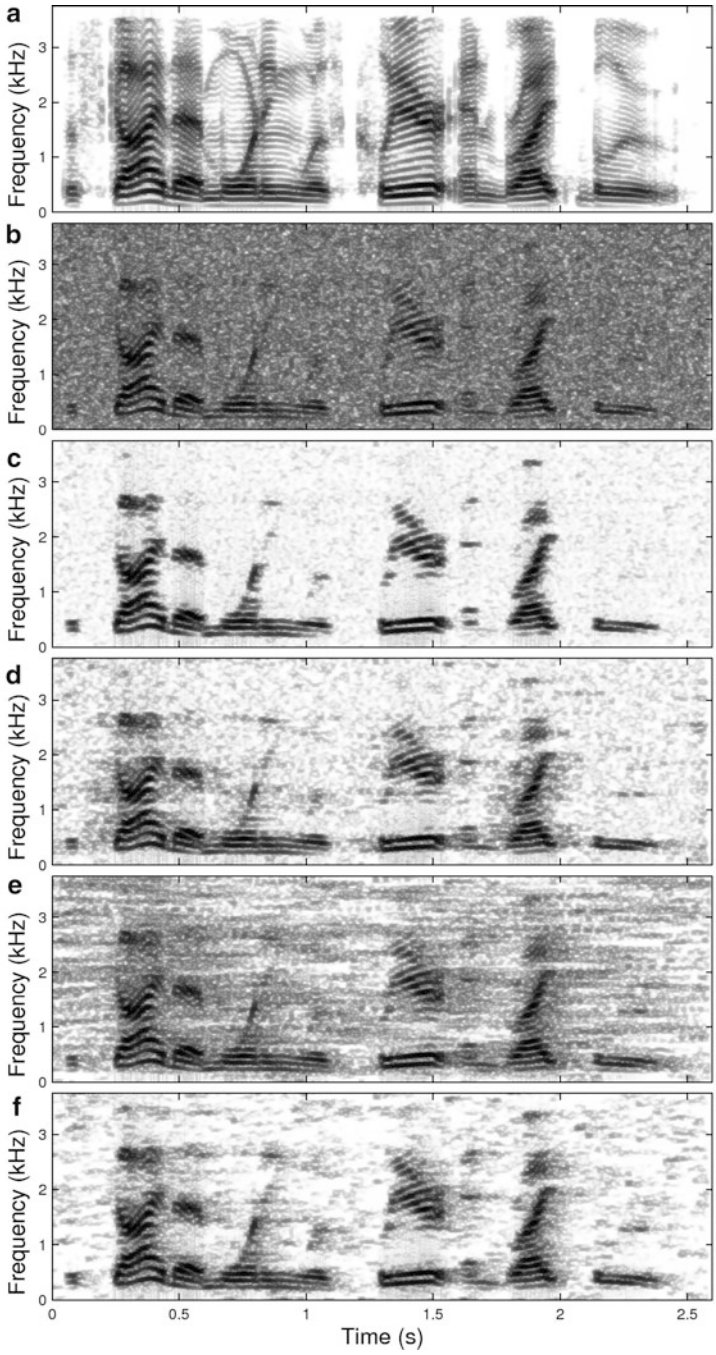

Fig. 10.8 Spectrograms of an utterance, "The sky that morning was clear and bright blue", by a male speaker from the Noizeus speech corpus: (**a**) clean speech; (**b**) speech degraded by AWGN at 5 dB SNR; and noisy speech enhanced using: (**c**) MME+SPU [42]; (**d**) ModSSub [44]; (**e**) ModWiener [42]; and (**f**) ModKalman [57]

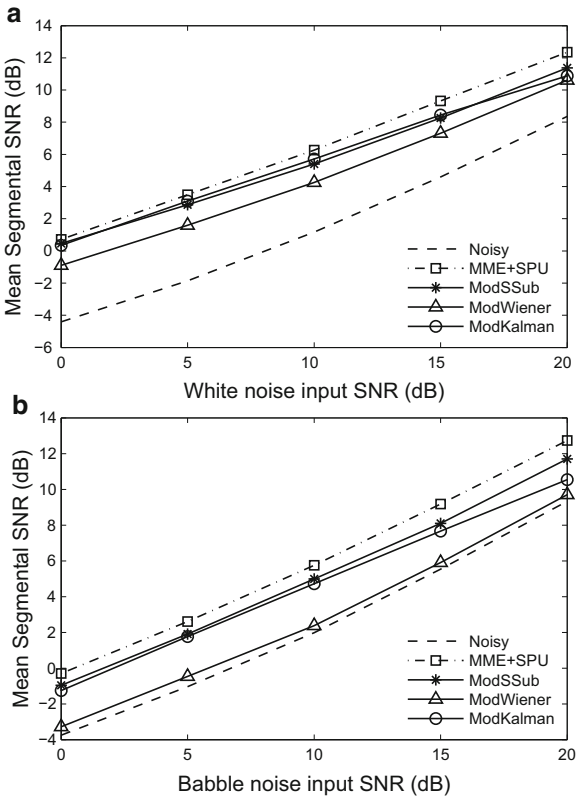

Fig. 10.9 Mean segmental SNR (dB) for: (1) noisy; and stimuli generated by processing noisy stimuli with the following treatment types: (2) MME+SPU [42]; (3) ModSSub [44]; (4) ModWiener [42]; and (5) ModKalman [57]. Plot (**a**) shows results for stimuli degraded with AWGN; and (**b**) for stimuli degraded with Babble noise

PESQ metric favours the more suppressed background noise of the ModSSub, and does not penalise the musical noise distortion present in ModSSub (and ModKalman) stimuli in the same way that human listeners typically do.

10.5 Conclusion

In this chapter, a review of speech enhancement methods which process the short-time modulation domain has been presented. These methods have utilised a short-time Fourier modulation AMS framework to modify the short-time modulation spectrum in order to improve the quality of speech degraded by additive noise distortion. This approach addresses limitations in previous modulation domain enhancement methods which predominantly applied filters designed from the long-term properties of the speech modulation spectrum, and ignoring the nonstationary properties of speech and noise.

Using the modulation AMS framework, noise was suppressed using a number of speech enhancement algorithms, including spectral subtraction ModSSub and

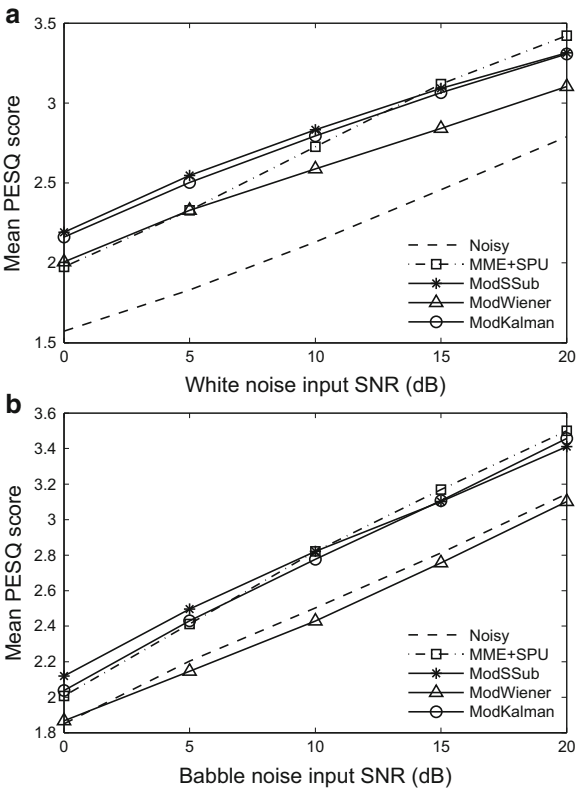

Fig. 10.10 Mean PESQ scores for: (1) noisy; and stimuli generated by processing noisy stimuli with the following treatment types: (2) MME+SPU [42]; (3) ModSSub [44]; (4) ModWiener [42]; and (5) ModKalman [57]. Plot (**a**) shows results for stimuli degraded with AWGN; and (**b**) for stimuli degraded with Babble noise

MMSE magnitude estimation (MME). ModSSub was found to improve the quality of processed stimuli in comparison to acoustic spectral subtraction and AME. ModSSub stimuli were found to have effective noise suppression (like spectral subtraction), but with considerably reduced introduction of musical noise for appropriately selected MFD. Thus, the MFD was an important parameter of the method, providing a trade-off between musical noise distortion (for shorter MFD) and the introduction of spectral smearing (for larger MFD), with longer MFD of around 256 ms providing the best compromise. This use of larger MFD also resulted in the need for longer silence regions for noise estimation and updates, making this approach less adaptive to changing noise conditions.

MMSE magnitude estimation in the modulation domain (MME) addressed this problem, resulting in further improvements in stimuli quality. As MMSE magnitude estimation does not introduce musical noise (for appropriately selected decision-directed smoothing parameter), a short MFD of 32 ms could be used. The result was stimuli which still improved background noise suppression compared to AME (though less effective than ModSSub), but without the introduction of spectral smearing observed in ModSSub, nor the musical noise of spectral subtraction and modulation domain Kalman filtering methods. However, MME also required a large

decision-directed a priori SNR estimation smoothing parameter, which results in slower updates of the noise estimate to changing noise properties, resulting in a smoothing of the spectrum, heard as a loss of crispness in speech. Some further improvement in quality was shown to be achievable through use of log-domain processing (LogMME) and particularly MME in the presence of speech uncertainty (MME+SPU). Common to both MME-based methods and ModSSub is the disadvantage of their additional computational complexity, but also the advantage when processing less stationary noise types in that subbands are processed independently. Comparisons of various modulation domain based methods, including modulation domain based Wiener filtering and Modulation domain Kalman filtering, indicated MME+SPU processed stimuli to be preferred over stimuli processed using other modulation domain based methods.

As a final comment, we note that here we have considered only the effect of modulation processing on speech quality. Results indicate that there is no improvement in intelligibility, and that for improved intelligibility the RI-modulation domain is more beneficial [53].

References

1. J. Allen, L. Rabiner, A unified approach to short-time Fourier analysis and synthesis. Proc. IEEE **65**(11), 1558–1564 (1977)
2. T. Arai, M. Pavel, H. Hermansky, C. Avendano, Intelligibility of speech with filtered time trajectories of spectral envelopes, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Philadelphia, PA, Oct 1996, pp. 2490–2493
3. L. Atlas, Modulation spectral transforms: application to speech separation and modification. Tech. Rep. 155. IEICE, University of Washington, Washington, WA (2003)
4. L. Atlas, S. Shamma, Joint acoustic and modulation frequency. EURASIP J. Appl. Signal Process. **2003**(7), 668–675 (2003)
5. L. Atlas, M. Vinton, Modulation frequency and efficient audio coding, in *Proceedings of the SPIE The International Society for Optical Engineering*, vol. 4474 (2001), pp. 1–8
6. S. Bacon, D. Grantham, Modulation masking: effects of modulation frequency, depth, and phase. J. Acoust. Soc. Am. **85**(6), 2575–2580 (1989)
7. M. Berouti, R. Schwartz, J. Makhoul, Enhancement of speech corrupted by acoustic noise, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4., Washington, DC, Apr 1979, pp. 208–211
8. S. Boll, Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acoust. Speech Signal Process. **27**(2), 113–120 (1979)
9. O. Cappe, Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor. IEEE Trans. Speech Audio Process. **2**(2), 345–349 (1994)
10. I. Cohen, Relaxed statistical model for speech enhancement and a priori SNR estimation. IEEE Trans. Speech Audio Process. **13**(5), 870–881 (2005)
11. D. Depireux, J. Simon, D. Klein, S. Shamma, Spectrotemporal response field characterization with dynamic ripples in ferrect primary auditory cortex. J. Neurophysiol. **85**(3), 1220–1234 (2001)
12. R. Drullman, J. Festen, R. Plomp, Effect of reducing slow temporal modulations on speech reception. J. Acoust. Soc. Am. **95**(5), 2670–2680 (1994)
13. R. Drullman, J. Festen, R. Plomp, Effect of temporal envelope smearing on speech reception. J. Acoust. Soc. Am. **95**(2), 1053–1064 (1994)

14. Y. Ephraim, D. Malah, Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. **32**(6), 1109–1121 (1984)
15. Y. Ephraim, D. Malah, Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. **33**(2), 443–445 (1985)
16. T. Falk, S. Stadler, W.B. Kleijn, W.-Y. Chan, Noise suppression based on extending a speech-dominated modulation band, in *Proceedings of the ISCA Conference of the International Speech Communication Association (INTERSPEECH)*, Antwerp, Aug 2007, pp. 970–973
17. R. Goldsworthy, J. Greenberg, Analysis of speech-based speech transmission index methods with implications for nonlinear operations. J. Acoust. Soc. Am. **116**(6), 3679–3689 (2004)
18. R. Gray, A. Buzo, A. Gray, Y. Matsuyama, Distortion measures for speech processing. IEEE Trans. Acoust. Speech Signal Process. **28**(4), 367–376 (1980)
19. S. Greenberg, T. Arai, The relation between speech intelligibility and the complex modulation spectrum, in *Proceedings of the ISCA European Conference on Speech Communication and Technology (EUROSPEECH)*, Aalborg, Sept 2001, pp. 473–476
20. D. Griffin, J. Lim, Signal estimation from modified short-time Fourier transform. IEEE Trans. Acoust. Speech Signal Process. **32**(2), 236–243 (1984)
21. H. Hermansky, N. Morgan, RASTA processing of speech. IEEE Trans. Speech Audio Process. **2**, 578–589 (1994)
22. H. Hermansky, E. Wan, C. Avendano, Speech enhancement based on temporal processing, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Detroit, MI, May 1995, pp. 405–408
23. T. Houtgast, H. Steeneken, A review of the MTF concept in room acoustics and its use for estimating speech intelligibility in auditoria. J. Acoust. Soc. Am. **77**(3), 1069–1077 (1985)
24. X. Huang, A. Acero, H. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development* (Prentice Hall, Upper Saddle River, 2001)
25. S. Kamath, P. Loizou, A multi-band spectral subtraction method for enhancing speech corrupted by colored noise, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2002)
26. N. Kanedera, T. Arai, H. Hermansky, M. Pavel, On the relative importance of various components of the modulation spectrum for automatic speech recognition. Speech Commun. **28**(1), 43–55 (1999)
27. D. Kim, A cue for objective speech quality estimation in temporal envelope representations. IEEE Signal Process. Lett. **11**(10), 849–852 (2004)
28. D. Kim, Anique: an auditory model for single-ended speech quality estimation. IEEE Trans. Speech Audio Process. **13**(5), 821–831 (2005)
29. B. Kingsbury, N. Morgan, S. Greenberg, Robust speech recognition using the modulation spectrogram. Speech Commun. **25**(1–3), 117–132 (1998)
30. T. Kinnunen, Joint acoustic-modulation frequency for speaker recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1. Toulouse, May 2006, pp. 665–668
31. T. Kinnunen, K. Lee, H. Li, Dimension reduction of the modulation spectrogram for speaker verification, in *Proceedings of ISCA Speaker and Language Recognition Workshop (ODYSSEY)*, Stellenbosch, Jan 2008
32. N. Kowalski, D. Depireux, S. Shamma, Analysis of dynamic spectra in ferret primary auditory cortex: I. Characteristics of single unit responses to moving ripple spectra. J. Neurophysiol. **76**(5), 3503–3523 (1996)
33. J. Lim, A. Oppenheim, Enhancement and bandwidth compression of noisy speech. Proc. IEEE **67**(12), 1586–1604 (1979)
34. P. Loizou, Speech Enhancement: Theory and Practice (Taylor and Francis, Boca Raton, 2007)
35. X. Lu, S. Matsuda, M. Unoki, S. Nakamura, Temporal contrast normalization and edge-preserved smoothing of temporal modulation structures of speech for robust speech recognition. Speech Commun. **52**(1), 1–11 (2010)

36. J. Lyons, K. Paliwal, Effect of compressing the dynamic range of the power spectrum in modulation filtering based speech enhancement, in *Proceedings of ISCA Conference of the International Speech Communication Association (INTERSPEECH)*, Brisbane, Sep 2008, pp. 387–390
37. N. Malayath, H. Hermansky, S. Kajarekar, B. Yegnanarayana, Data-driven temporal filters and alternatives to GMM in speaker verification. Digit. Signal Proces. **10**(1–3), 55–74 (2000)
38. R. McAulay, M. Malpass, Speech enhancement using a soft-decision noise suppression filter. IEEE Trans. Acoust. Speech Signal Process. **28**(2), 137–145 (1980)
39. N. Mesgarani, S. Shamma, Speech enhancement based on filtering the spectrotemporal modulations, in *Proceedings of IEEE International Conference Acoustics Speech and Signal Processing (ICASSP)*, vol. 1, Philadelphia, PA, Mar 2005, pp. 1105–1108
40. C. Nadeu, P. Pachés-Leal, B.-H. Juang, Filtering the time sequences of spectral parameters for speech recognition. Speech Commun. **22**(4), 315–332 (1997)
41. K. Paliwal, B. Schwerin, K. Wójcicki, Role of modulation magnitude and phase spectrum towards speech intelligibility. Speech Commun. **53**(3), 327–339 (2011)
42. K. Paliwal, B. Schwerin, K. Wójcicki, Speech enhancement using minimum mean-square error short-time spectral modulation magnitude estimator. Speech Commun. **54**(2), 282–305 (2012)
43. K. Paliwal, K. Wójcicki, Effect of analysis window duration on speech intelligibility. IEEE Signal Process. Lett. **15**, 785–788 (2008)
44. K. Paliwal, K. Wójcicki, B. Schwerin, Single-channel speech enhancement using spectral subtraction in the short-time modulation domain. Speech Commun. **52**(5), 450–475 (2010)
45. K. Payton, L. Braida, A method to determine the speech transmission index from speech waveforms. J. Acoust. Soc. Am. **106**(6), 3637–3648 (1999)
46. J. Picone, Signal modeling techniques in speech recognition. Proc. IEEE **81**(9), 1215–1247 (1993)
47. S. Quackenbush, T. Barnwell, M. Clements, *Objective Measures of Speech Quality* (Prentice Hall, Englewood Cliffs, 1988)
48. T. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice* (Prentice Hall, Upper Saddle River, 2002)
49. L. Rabiner, R. Schafer, *Theory and Applications of Digital Speech Processing* (Pearson Higher Education, Upper Saddle River, 2011)
50. A. Rix, J. Beerends, M. Hollier, A. Hekstra, Perceptual Evaluation of Speech Quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs. ITU-T Recommendation P.862 (2001)
51. P. Scalart, J. Filho, Speech enhancement based on a priori signal to noise estimation, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Process (ICASSP)*, vol. 2. Atlanta, GA, May 1996, pp. 629–632
52. C. Schreiner, J. Urbas, Representation of amplitude modulation in the auditory cortex of the cat: I. The anterior auditory field (AAF). Hear. Res. **21**(3), 227–241 (1986)
53. B. Schwerin, K. Paliwal, Using STFT real and imaginary parts of modulation signals for MMSE-based speech enhancement. Speech Commun. **58**, 49–68 (2014)
54. S. Shamma, Auditory cortical representation of complex acoustic spectra as inferred from the ripple analysis method. Netw. Comput. Neural Syst. **7**(3), 439–476 (1996)
55. B. Shannon, K. Paliwal, Role of phase estimation in speech enhancement, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Pittsburgh, PA, Sep 2006, pp. 1423–1426
56. S. Sheft, W. Yost, Temporal integration in amplitude modulation detection. J. Acoust. Soc. Am. **88**(2), 796–805 (1990)
57. S. So, K. Paliwal, Modulation-domain Kalman filtering for single-channel speech enhancement. Speech Commun. **53**(6), 818–829 (2011)
58. J. Sohn, N.S. Kim, W. Sung, A statistical model-based voice activity detection. IEEE Signal Process. Lett. **6**(1), 1–3 (1999)
59. H. Steeneken, T. Houtgast, A physical method for measuring speech-transmission quality. J. Acoust. Soc. Am. **67**(1), 318–326 (1980)

60. J. Thompson, L. Atlas, A non-uniform modulation transform for audio coding with increased time resolution, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Process (ICASSP)*, vol. 5, Hong Kong, Apr 2003, pp. 397–400
61. V. Tyagi, I. McCowan, H. Misra, H. Bourland, Mel-cepstrum modulation spectrum (MCMS) features for robust ASR, in *Proceedings of IEEE Workshop Automatic Speech Recognition and Understanding (ASRU)*, St. Thomas, VI, Dec 2003
62. P. Vary, R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment* (Wiley, West Sussex, 2006)
63. N. Virag, Single channel speech enhancement based on masking properties of the human auditory system. IEEE Trans. Speech Audio Process. **7**(2), 126–137 (1999)
64. S.V. Vuuren, H. Hermanshy, On the importance of components of the modulation spectrum for speaker verification, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, vol. 7, Sydney, Nov 1998, pp. 3205–3208
65. D. Wang, J. Lim, The unimportance of phase in speech enhancement. IEEE Trans. Acoust. Speech Signal Process. **30**(4), 679–681 (1982)
66. X. Xiao, E. Chng, H. Li, Normalization of the speech modulation spectra for robust speech recognition, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Process (ICASSP)*, vol. 4, Monolulu, HI, Apr 2007, pp. 1021–1024