Autoregressive models for text independent speaker identification in noisy environments

by

Moataz El Ayadi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The closed-set speaker identification problem is defined as the search within a set of persons for the speaker of a certain utterance. It is reported that the Gaussian mixture model (GMM) classifier achieves very high classification accuracies (in the range 95% - 100%) when both the training and testing utterances are recorded in sound proof studio, i.e., there is neither additive noise nor spectral distortion to the speech signals.

However, in real life applications, speech is usually corrupted by noise and band-limitation. Moreover, there is a mismatch between the recording conditions of the training and testing environments. As a result, the classification accuracy of GMM-based systems deteriorates significantly. In this thesis, we propose a two-step procedure for improving the speaker identification performance under noisy environment. In the first step, we introduce a new classifier: vector autoregressive Gaussian mixture (VARGM) model. Unlike the GMM, the new classifier models correlations between successive feature vectors. We also integrate the proposed method into the framework of the universal background model (UBM). In addition, we develop the learning procedure according to the maximum likelihood (ML) criterion. Based on a thorough experimental evaluation, the proposed method achieves an improvement of 3 to 5% in the identification accuracy.

In the second step, we propose a new compensation technique based on the generalized maximum likelihood (GML) decision rule. In particular, we assume a general form for the distribution of the noise-corrupted utterances, which contains two types of parameters: clean speech-related parameters and noise-related parameters. While the clean speech related parameters are estimated during the training phase, the noise related parameters are estimated from the corrupted speech in the testing phase. We applied the proposed method to utterances of 50 speakers selected from the TIMIT database, artificially corrupted by convolutive and additive noise. The signal to noise ratio (SNR) varies from 0 to 20 dB. Simulation results reveal that the proposed method achieves good robustness against variation in the

SNR. For utterances corrupted by covolutive noise, the improvement in the classification accuracy ranges from 70% for SNR = 0 dB to around 4% for SNR = 10dB, compared to the standard ML decision rule. For utterances corrupted by additive noise, the improvement in the classification accuracy ranges from 1% to 10% for SNRs ranging from 0 to 20 dB.

The proposed VARGM classifier is also applied to the speech emotion classification problem. In particular, we use the Berlin emotional speech database to validate the classification performance of the proposed VARGM classifier. The proposed technique provides a classification accuracy of 76% versus 71% for the hidden Markov model, 67% for the k-nearest neighbors, 55% for feed-forward neural networks. The model gives also better discrimination between high-arousal emotions (joy, anger, fear), low arousal emotions (sadness, boredom), and neutral emotions than the HMM.

Another interesting application of the VARGM model is the blind equalization of multi input multiple output (MIMO) communication channels. Based on VARGM modeling of MIMO channels, we propose a four-step equalization procedure. First, the received data vectors are fitted into a VARGM model using the expectation maximization (EM) algorithm. The constructed VARGM model is then used to filter the received data. A Baysian decision rule is then applied to identify the transmitted symbols up to a permutation and phase ambiguities, which are finally resolved using a small training sequence. Moreover, we propose a fast and easily implementable model order selection technique. The new equalization algorithm is compared to the whitening method and found to provide less symbol error probability. The proposed technique is also applied to frequency-flat slow fading channels and found to provide a more accurate estimate of the channel response than that provided by the blind de-convolution exploiting channel encoding (BDCC) method and at a higher information rate.

Acknowledgements

I gratefully acknowledge my indebtedness to my supervisors Prof. Mohamed Kamel and Prof. Fakhri Karray who initiated and supervised this work. They conducted a lot of fruitful discussions and have been a source of inspiration and encouragement. They provided me with a lot of references and materials that were very essentials for the completion of this work. Moreover, they enrolled me in other research activities that enriched my experience in speech recognition.

I would like to express my gratefulness to all my friends who supported me during my PhD program. They helped me overcome some stressful situations and pass both the comprehensive exam and the final defense easily. In particular, I would like to thank Ahmed Shamy, Azmy Faisal, Ayman Abdel Rahman, Hatem Al-Beheiry, Hatem Zein El Din, Nizar Abdullah, and Rami Langar. Special thanks go to Khaled Hammouda and Shady Shehata. They helped me a lot in technical issues and provided me with a convenient research environment. I am also very grateful to Ahmed Khairy, Mohamed Hammouda, and Shady Shehata for their encouragement and help during my last year of the program. They also helped me in the preparation of the presentation of the final defense.

I am also very thankful to my PAMI colleagues who were keen to attend my presentations and provide me with their useful comments and feedback. In particular, I would like to thank Abbas Ahmadi, Sami Ullah, Akram Al Ghazali, Hanan Ayad, Mostafa Adel, Nabil Drawil, and Rasha Kashef. I wish for them all the best and success in their future life.

Finally, I do not find words to express my deep gratefulness and love to my father, Prof. Hatem El Ayadi, my mother, Mona, my brother, Moatase, and my sister, MennatAllah, who supported me in many aspects and prayed for me a lot to pass my exams easily.

Dedication

To my parents, my brother, and my sister.

Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

AIC Akaike information criterion

ANN artificial neural networks

AWGM additive white Gaussian noise

BDCC blind de-convolution exploiting channel coding

BER bit error probability

BIC Bayesian information criterion

CMS cpestral mean normalization

CMVN cepstral mean and variance normalization

CSI channel state information

CSLU center for spoken language understanding

DTW dynamic time warping

EM expectation maximization

GLRT generalized likelihood ratio test

GML generalized maximum likelihood

GMM Gaussian mixture model

GVQ group vector quantization

HMM hidden Markov model

HOS higher order statistics

HTK hidden Markov toolkit

IFC inverse filter criteria

IIR infinite impulse response

KIC Kullback information criterion

kNN k-nearest neighbor

LBG-VQ Linde-Buzo-Gray vector quantization

LDPC low density parity check

LVQ learning vector quantization

MAP maximum a posteriori

MCE minimum classification error

MFCC mel-frequency cepstrum coefficients

MIMO multiple input multiple output

ML maximum likelihood

MLP multi-layer perceptron

MMI maximum mutual information

NAR-HMM non-stationary autoregressive hidden Markov model

NMSE normalized mean square error

OLS ordinary least squares

PMC parallel model combination

QPSK quadrature phase shift keying

RBF radial basis function

SE super exponential

SISO single input single output

SMS speaker model synthesis

SNR signal to noise ratio

SOS second-order statistics

SRM structural risk minimization

SS spectral substraction

SVM support vector machines

UBM universal background model

VAR vector autoregressive

VARGM vector autoregressive Gaussian mixture

VMA vector moving average

VQ vector quantization

Chapter 1

Introduction

The speech signal is the fastest and the most natural way of communication between humans. Moreover, it carries several types of information. From the speech point of view, it carries the following types of information: linguistic information (the spoken word sequence), speaker information (e.g. identity, emotional state, accent), and environmental information (e.g. the signal to noise ratio and the transmission bandwidth). Such nice properties of the speech signal have motivated researchers to think of speech as a fast and efficient way of interaction between human and machine. However, this requires that the machine should have the sufficient intelligence to *recognize* human voices. This faculty is referred to as Voice Recognition to which we generally attribute the faculties of *Speech Recognition* and *Speaker Recognition*.

Speech recognition is defined as the process of extracting the spoken words and phrases from a given speech utterances. It has many applications such as voice dialing, call routing, content-based spoken audio search, simple data entry, preparation of structured documents, and speech-to-text processing. On the other hand, the research on speaker recognition is concerned with extracting the identity of the person speaking the utterance. Some of the important applications of speaker recognition include customer verification for bank transactions, access to bank accounts through telephones, control on the use of credit cards, machine-voice commands

and security check in military environments [17].

The first speaker recognition system was implemented at Bell labs in the late 60's by Lawrence Kersta [67]. The basic idea of that system is based on the visual comparison between the spectrogram of the testing system and those of the training candidates. Over the past four decades, a significant progress has been achieved in speaker recognition. However, *natural* speaker recognition is still a difficult task due to many factors such as mismatch between the training and testing recording conditions (e.g. different microphones for enrollment and verification), different levels of surrounding noise, spectral distortion of speech caused by the band-limitation of the communication medium (e.g. the telephone channel), and multi-path fading effects [79, 17]. In this thesis, we mainly address the speaker recognition problem in noisy envirnments.

1.1 Speaker recognition: principles and applications

The research on speaker recognition is divided into three main categories: *identification*, *verification*, and *segmentation* [42, 18, 98].

The speaker identification problem is defined as the determination of a speaker identity from his/her voice. A speaker identification system is said to be *open-set* if it can determine whether the given testing utterance belongs to the set of enrolled speakers or not. Otherwise, it is called a *closed-set* speaker identification system [17]. Another distinguishing feature of speaker recognition system is whether it is text-dependent or text-independent. In text-dependent systems, the underlying texts of training and testing utterances are the same. On the other hand, the task is more difficult in text-independent systems where there is no restriction on the sentences spoken by the user of the system [94].

In speaker verification, the goal is to decide whether a certain speech utterance belongs to a certain speaker or not [36]. Therefore, it is a binary decision problem.

This problem is also called speaker detection, speaker authentication, talker verification or authentication, and voice verification [62]. Usually, two kinds of speakers are defined for the speaker verification problem: *target* speakers, which refer to the normal users of the system, and *imposter* speakers, which refer to unwanted people who fake the voices of the target speakers. Therefore, the speaker verification problem is an open-set problem. Clearly, voice-stamp security applications are based on speaker verification. Furthermore, speaker verification is the basis of many practical applications.

In most speech recognition and speaker recognition systems, it is often assumed that the spoken utterance contains speech from one speaker only. However, in some applications, the voice of the intended speaker may be mixed with other speakers, e.g. a telephone conversation. In this case, it is necessary to *divide* the speech utterance into segments of each speaker in the conversation before applying the speaker recognition techniques. This task is called *speaker segmentation and clustering*. It is important in applications involving multi-speakers conversations such as meetings and TV shows.

1.2 Thesis motivation and contributions

In this thesis, we mainly consider the text-independent closed-set speaker identification problem in real-life environment. When both the training and testing utterances are recorded in clean environment, e.g. sound-proof studio, very high recognition accuracies (in the range %95 - %100) can be achieved using the Gaussian mixture model (GMM) classifier [96, 103]. However, in real life applications, there are many factors that significantly degrades the classification performance [60, 48] such as:

- the use of different microphones for enrolment and verification.
- the surrounding noise. When the signal to noise ratio differs from one utterance to another, the degradation in performance is more severe.

- the spectral distortion of the testing utterances caused by transmitting the speech signal through a band-limited channel.
- multi-path fading effects [79, 17].
- extreme emotional states of the speaker, e.g. stress or happiness.
- sickness and aging [17].

Therefore, it is still difficult to implement an accurate speaker recognition system in practice [79]. These factors motivated research on how to reduce the effect of handset/channel mismatch. Channel compensation techniques can be categorized into three groups: feature-based methods, e.g. spectral subtraction (SS) [14], cepstral mean subtraction or RASTA, model-based methods, e.g. speaker model synthesis [110] and parallel model combination (PMC) [43], and score-based methods, e.g. H-Norm [100], Z-Norm [9], and T-Norm [5]. A brief review on these compensation methods is given in Chapter 2.

The main contribution in this thesis is the development of a two-step procedure for improving the classification performance of GMM-based text-independent speaker identification systems. In the first step of our procedure, we relax the assumption of statistical independence between successive feature vectors, employed in the ordinary GMM-based classification framework [96]. Although this assumption is incorrect, the GMM classifier provides high classification accuracies in clean environments [103]. However, we believe that modeling correlations between feature vectors is useful for utterances recorded in telephone channels. The main reason is that the telephone channel can be modeled by a bandpass filter, which naturally introduces correlation between successive speech time samples. It is also believed that modeling speaker-dependent temporal information present in the prompted phrases is useful in speaker identification [17, 130].

The correlation between successive feature vectors is modeled through an autoregressive relation. Therefore, the proposed model is a generalization to the

standard vector autoregressive (VAR) model in which the distribution of the innovation sequence is a mixture of Gaussian densities. The new introduced model is called vector autoregressive Gaussian mixture (VARGM) model. It can be also considered as a combination of the standard VAR (modeling correlation between feature vectors) and the standard GMM since it models the multi-modality in the distribution of the training data. When applied to the 2000 NIST speaker recognition evaluation, the proposed VARGM model is shown to provide a 3-5% increase in the classification accuracy over the standard GMM-based systems.

In the second phase in our improvement procedure, we attempt to overcome the problem of the mismatch between the recording environments of the training and the testing utterances. Inspired by the successful application of the generalized likelihood ratio test (GLRT) in radar and sonar signal detection [12] and in voice/unvoiced detection [40], we modified the GLRT to fit into the multi-hypotheses classification problems. The new introduced rule is called the generalized maximum likelihood (GML) decision rule. We applied the proposed method to utterances in the TIMIT database, artificially corrupted by convolutive and additive noise. The signal to noise ratio (SNR) varies from 0 to 20 dB. Experiments were applied for 50 speakers. Simulation results reveal that the proposed method achieves good robustness against variation in the signal to noise ratio.

As a side application, we successfully applied our proposed VARGM model to the speech emotion classification problem [6]. When applied to the Berlin emotional speech database [15], the proposed technique improves the classification accuracy by 5% over the hidden Markov model (HMM), 9% over the k-nearest neighbors (k-NN), and 21% over the feed-forward artificial neural networks (ANN). The model gives also better discrimination between high-arousal emotions (joy, anger, fear), low arousal emotions (sadness, boredom), and neutral emotions than the HMM.

Finally, the proposed GML adaptation framework is modified to fit into the problem of blind equalization of multi input multiple output (MIMO) communication channels. The main motivations behind considering this application are the

great similarities between the two problems and the recent interest in the latter problem [117, 116, 11, 81, 65, 24]. It should be mentioned that the scalar autoregressive Gaussian mixture model was introduced and proposed to blindly equalize single input single output (SISO) channels in [120]. However, besides considering the MIMO case, we generalize their approach in two other ways. First, complex time series are considered instead of real ones. This enables us to deal with the complex baseband representation of modulated signals. Furthermore, and unlike [120], we consider the problem of estimating the channel state information (CSI). The new equalization algorithm is compared to the whitening method [114] and found to provide less symbol error probability. It is also applied to frequency-flat slow fading channels and found to provide a more accurate estimate of the channel response than that provided by the blind de-convolution exploiting channel coding (BDCC) method and at a higher information rate.

1.3 Thesis organization

The thesis consists of seven chapters, the first of which is the introduction. We conclude this chapter with a description of the mathematical notations used throughout the thesis.

In chapter 2, a brief review of the text-independent speaker identification systems is given. Basically, we give a qualitative formulation for the problem, describe a generic structure of a speaker identification system, and review the most common and prominent feature extraction and classification techniques. We conclude the chapter by a brief survey of mismatch reduction techniques.

Chapter 3 covers the basic theory of GMMs: definition and parameter estimation using the expectation maximization (EM) algorithm [29]. It also describes in details two prominent classification frameworks employing GMMs as the core statistical classifiers: the standard ML framework and the Gaussian mixture model/universal background model (GMM/UBM) framework.

In Chapter 4, we present the theory of the proposed VARGM classifier. We start by briefly reviewing the simple VAR model and how to extend it to our proposed VARGM model. We then consider the parameter estimation problem and the model order selection problem for the VARGM model. In the last section of this chapter, the two classification frameworks addressed in chapter 2 are reconsidered again but with the VARGM model as the core statistical classifier.

In chapter 5, the GML-based adaptation framework is described. Basically, we illustrate the adaptation architecture and discuss the parameter estimation and the model order selection problems. We also address the proposed application of blind equalization of MIMO channels in this chapter. In particular, we mathematically formulate the equalization problem and describe in details our proposed equalization procedure.

The simulation results of all the above-mentioned techniques and suggested applications are combined in chapter 6. We also include our experiments with the speech emotion classification problem in this chapter.

Finally, important conclusions and possible extensions to this work are stated in Chapter 7.

1.4 Notations

In this thesis, italic letters are used to represent scalars or sets while lower case bold letters represent vectors. For matrices, upper case bold letters are used. There is no distinction in notation between deterministic and random variables as this will be understood from the context.

For iterative algorithms, a superscript with parenthesis indicates the iteration number. For example, $\mathbf{A}^{(s)}$ denotes the s^{th}-iterated value of $\mathbf{A}$.

Sequences may be represented in one of three ways: using braces with ... inside, e.g. $\{\mathbf{x}[1], \ldots, \mathbf{x}[N]\}$, using braces with a lower limit and an upper limits on the

closing braces, e.g. $\{\mathbf{x}[n]\}_{n=1}^{N}$, or using a colon between the starting index and the ending index, $\mathbf{x}[1:N]$.

The probability of a certain event will be denoted by $P(.)$ while the symbol p() is used with probability density functions. That is, if x is a random variable, then p(x) denotes its probability function. If y is another random variable, then p($x|y$) denotes the conditional probability density function of x given y. The notation p($x|\lambda$), where λ is a deterministic variable, means that the probability density function of x depends on parameter λ. The symbol $E\{\}$ means expectation. The multivariate normal probability density function with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is denoted by $\mathbb{N}(;\boldsymbol{\mu},\boldsymbol{\Sigma})$, i.e.,

$$\mathbb{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}\,|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

where D is the dimensionality of the vector $\mathbf{x}$.

It is always assumed that all time series are causal, i.e., their values are equal to zero at non-positive time instants. Therefore, summations like $\sum_{n=1}^{N}\mathbf{x}[n-i]$, $i>0$ should cause no ambiguity because the first i terms of this series are zeros.

Finally, all over the thesis, the following variables are used with fixed interpretation

1. n denotes an index of a time sample of feature vectors,
2. d denotes a specific dimension,
3. m denotes an index of Gaussian components,
4. p denotes an index of autoregression matrices.

Chapter 2

Text-independent speaker identification: a brief review

Speaker recognition refers to the process of extracting information about the speaker from his/her voice. Figure 2.1 illustrates a typical architecture of a speaker identification system. The analog speech signal is filtered and then converted into a digital signal. Since the task is to identify the person talking rather than what the person is saying, the speech signal must be processed to extract measures of speaker variability instead of segmental features. Although there are no exclusively speaker distinguishing features, features based on the spectral analysis of the speech signal are known to be powerful in speaker recognition [96]. In particular, the mel-frequency cpestrum coefficients (MFCC) have been a typical choice for speaker recognition tasks because of their inherent robustness to noise and their ability to reflect the human perception of sounds [92]. Therefore, we primarily consider MFCC in this review. The last step in the speaker identification system is to match the extracted feature vectors with the stored speaker models, obtained in the training phase. The identified speaker is the one whose model gives the best match with the extracted testing feature vectors.

As mentioned in the introduction, in real life applications, there are some factors that cause a significant deterioration to the classification performance such

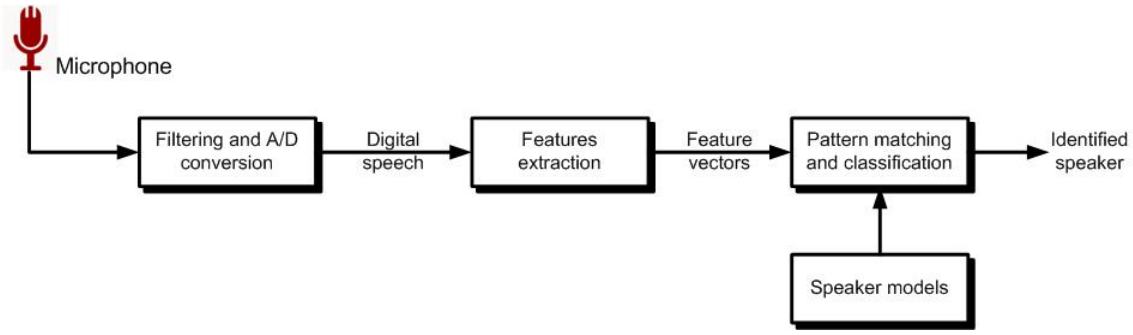

Figure 2.1: Functional block diagram of a speaker identification system.

as the surrounding noise and the spectral distortion caused by the communication channels. In this chapter, we present a brief survey on the classification methods as well as the compensation techniques used for closed-set text-independent speaker identification.

This chapter contains four sections. In section 2.1, we present a qualitative description of the closed-set text-independent speaker identification problem. In section 2.2, we discuss in some details the process of extracting the MFCC features from the speech signal. A quick review on popular classification techniques used in the context of speaker identification is given in section 2.3. Common compensation techniques are covered in section 2.4.

2.1 Problem formulation

Based on the above definition of the speaker identification problem, the closed-set speaker identification is merely a multi-class pattern recognition problem: the class labels correspond to the speakers' identities and the training and testing examples are the feature vectors extracted from the training and the testing utterances, respectively. Similar to all pattern classification problems, the speaker identification problem consists of two phases: *learning* (training) and *classification* (testing). In the learning phase, we have one or more speech utterances for each speaker to be enrolled in the system. The main objective in this phase is the construction of

a classifier that models the relevant characteristics of all speakers in the system. The available training speech together with their labels are used to estimate the classifier parameters.

In the classification phase, we have a sequence of feature vectors $\{\mathbf{x}[1], \ldots, \mathbf{x}[N]\}$, extracted from a testing speech utterance with unknown speaker identity. The main objective in this phase is the determination of the speaker that most likely uttered the given testing speech. The closed-set speaker identification problem is thus formulated as the following multi-hypotheses problem:

$$\mathcal{H}_i: \text{ The sequence } \{\mathbf{x}[1], \ldots, \mathbf{x}[N]\} \text{ is produced by speaker } i. \quad i = 1, \ldots, S,$$

where S is the number of speakers. If Ω_i is the decision region for the i^{th} speaker, the sets $\Omega_1, \ldots, \Omega_S$ are disjoint and exhaustive for the closed-set speaker identification problem, i.e., the union of the decision regions comprise the entire feature space. Thus, the classification system is forced to make one and only one decision for each incoming test utterances. It should be mentioned that there are other decision systems that allow the speaker identification system to reject the incoming testing signal or output more than one hypothesis such as the erasure decoding and the list decoding [50]. For more details about such decision strategies, the reader is referred to [102].

2.2 Feature extraction

The underlying assumption in most speech processing schemes is that the properties of a speech signal vary relatively slowly with time [93]. This leads us to the basic principle of speech analysis in which the speech signal is divided into short segments called *frames*. The time samples of each frame may be filtered and multiplied by a shaping window in order to enhance the spectral properties of the speech signal. Nonetheless, the speech time samples are rarely used as a representation in speaker recognition applications because they carry little information about the conveyed speaker [96]. Usually, spectral features are calculated from the speech samples of

each frame and combined into one vector. This vector is called the *feature vector* and is used to represent the corresponding speech frame. The feature extraction process is illustrated in figure 2.2.

The samples of each frame can be considered as the output of a linear time invariant system excited properly. The problem of speech analysis is to estimate the parameters of the linear time system producing each frame. Since the excitation and the impulse response of a linear time invariant system are related in a convolutional manner, the problem of speech analysis can be viewed as a problem in separating the component of a convolution. For that purpose, a complex cepstrum of a signal is defined as the inverse Fourier Transform of the logarithm of the signal spectrum. Formally, the cepstrum of a signal ,s_t, is given by

$$c_t = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |S\left(e^{j\omega}\right)| e^{j\omega t} d\omega, \tag{2.1}$$

where $|S\left(e^{j\omega}\right)|$ is the Fourier transform of the speech signal, i.e.,

$$S\left(e^{j\omega}\right) = \sum_{t=-\infty}^{\infty} s_t e^{-j\omega t}, \tag{2.2}$$

An interesting property for the cepstrum is that the cepstrum of the discrete time convolution of two signals equals to the summation of the cepstra of the individual signals. Thus, the cepstrum of each speech frame can be viewed as a superposition of the cepstra of the excitation and the impulse response of the speech model.

However, the ordinary cepstrum has two disadvantages. The first one is that the cepstrum is of infinite extent even when the original signal is of a finite duration. Although the cepstrum is a rapidly decaying function, a relatively large number of cepstral samples has to be extracted from each frame for an accurate representation of the cepstrum. This increases the computational requirements of the training and the testing algorithms. Another disadvantage is that the ordinary cepstrum does not adequately model the human perception to the frequency content of sounds.

Psychological studies show that the human perception to either pure tones or speech signals does not follow a linear scale [92]. This research has led to the idea

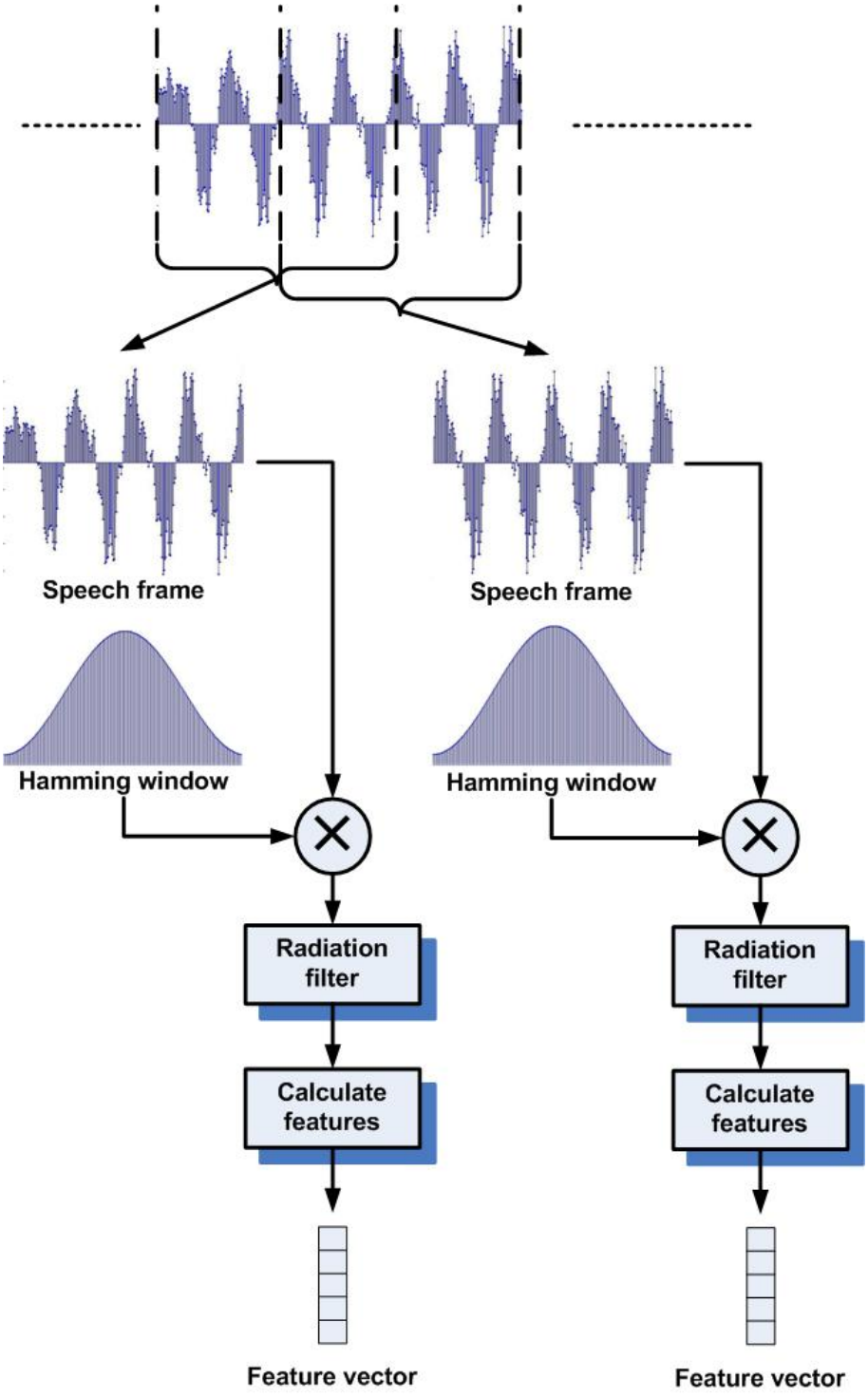

Figure 2.2: The speech feature extraction process.

of defining a subjective pitch of pure tones. Thus for each tone with an actual frequency, f, measured in Hz, a subjective pitch is measured on a scale called the

mel scale. As a reference point, the pitch of a 1 KHz tone, 40 dB above perceptual hearing threshold, is defined as 1000 mels. An empirical relation between the linear frequency (measured in Hz) and the mel frequency (measured in mels) is given by [28]

$$\text{mel}(f) = 1000\frac{\log(1 + f/700)}{\log(1 + 1000/700)}. \tag{2.3}$$

It can be noticed from the above formula that the relation between the mel frequency and the linear frequency is almost linear for low frequencies (below 700 Hz) and logarithmic for high linear frequencies (beyond 1KHz).

Another important subjective criterion of the frequency contents of a signal is the critical band that refers to the bandwidth at which subjective responses, such as loudness, become significantly different. The loudness of a band of noise at a constant sound pressure remains constant as the noise bandwidth increases up to the bandwidth of the critical band. After that, an increased loudness is perceived. Similarly, a subcritical bandwidth complex sound (multi-tone) of constant intensity is about as loud as equally intense pure tone of a frequency lying at the center of the band, regardless of the overall frequency separation of the multiple tones. When the separation exceeds the critical bandwidth, the complex sound is perceived as becoming louder.

One approach to simulating the above two subjective criteria is through the use of a bank of filters spaced uniformly on the warped mel frequency scale [25]. The modified cepstrum of $S(e^{j\omega})$ thus consists of the output power of these filters when $S(e^{j\omega})$ is input. Denoting these power coefficient by , $\tilde{S}_k, k = 1, \ldots, K$, we can calculate what is called the mel-frequency cepstrum, x_d, as

$$x_d = \sum_{k=1}^{K} \log(\tilde{S}_k) \cos\left(d(k - \frac{1}{2})\frac{\pi}{K}\right) \qquad d = 1, \ldots, D, \tag{2.4}$$

where D is the desired length of the cepstrum. Fig. 2.3 shows the frequency response magnitude of the filter bank used in our experimental study. Cepstral analysis is performed only over the telephone passband (300-3300 Hz). Each filter has a triangular bandpass frequency response, and the spacing as well as the

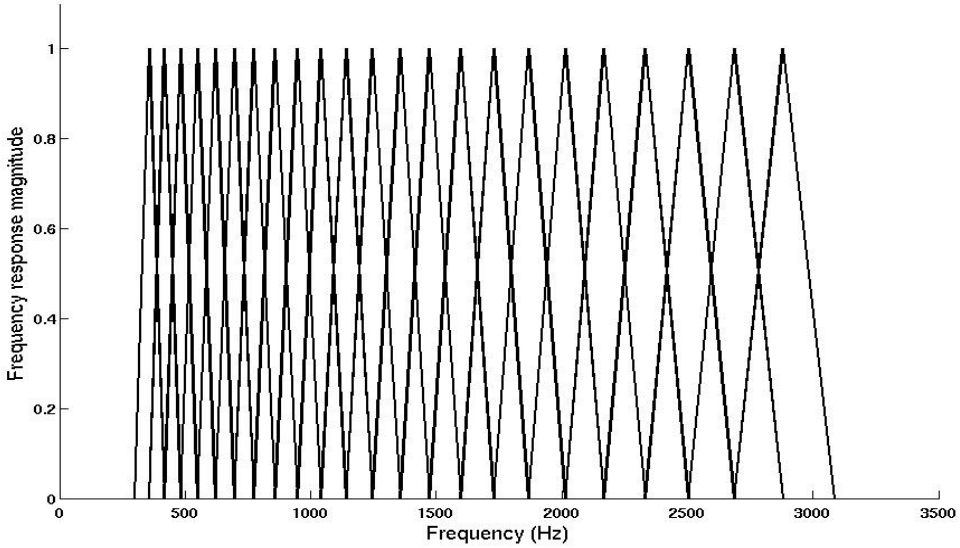

Figure 2.3: The filter bank design used in our experimental study. Each filter has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined by a constant mel frequency interval. (The spacing is approximately 62.5 mels and the width of the triangle is about 125 mels).

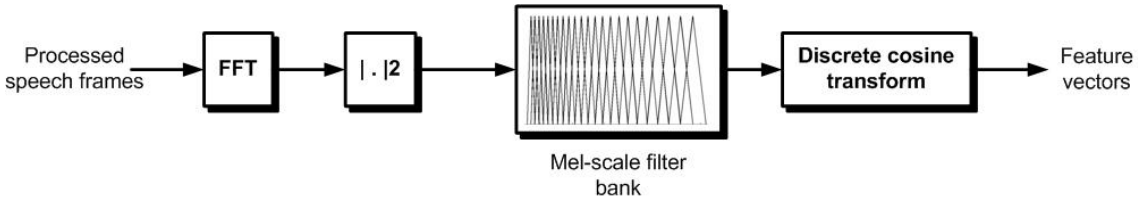

Figure 2.4: Functional block diagram of MFCC feature extraction.

bandwidth is determined by a constant mel frequency interval. (The spacing is approximately 55 mels and the width of the triangle is about 110 mels). A block diagram illustrating the complete procedure for extracting MFCCs from a speech signal is depicted in Fig. 2.4.

2.3 Classification techniques

Classification refers to deciding the class label (the unknown identity) of the testing signal. In closed-set speaker identification systems, classification is performed by assigning a score for each class that attempts to measure how likely the corresponding speaker produced the given testing utterance. A decision is made in favor of the speaker whose model provides the highest matching score. The classifier performance is measured by its ability to predict the true labels of unknown testing utterances as well as the time need for making a decision. In order to have a good classification performance, training examples (utterances) with known labels are used to estimate the classifier parameters.

Basically, there are two main approaches for learning. In the first approach, a model is constructed for each speaker. The training examples (feature vectors) of each speaker are used to train his corresponding model only. Thus, at the end of the training phase, we have S trained models; each is trained to exactly one of the speakers in the systems. In the testing phase, each model calculates a *likelihood* score with respect to the given testing utterances. This approach is sometimes called *unsupervised* learning [94] because, when each speaker model is trained, the corresponding class label information is not used. Examples of unsupervised modeling approaches include k-NN, vector quantization (VQ), GMM and HMM.

On the other hand, the *supervised* training approach refers to classification schemes that use all the training data of all speakers together with their corresponding labels to train the classifier. In the training phase, the classifier learns how to distinguish between different classes rather than learning each class alone. Examples of supervised classifiers include multi-layer perceptron (MLP) neural networks, radial basis functions (RBF) neural networks, support vector machines (SVM). In these classification techniques, a single classifier model is assigned to all classes and used for both training and classification. Another alternate configuration is to assign a model to each class like the unsupervised learning approach. However, each model is trained to favor its corresponding training data and unfavor the training

data of other speakers. The parameter estimation criterion in the latter configuration are said to be *discriminative*. In the context of speech recognition, popular discriminative estimation criteria include the minimum classification error (MCE) criterion [64] and the maximum mutual information (MMI) criterion [8]. Supervised algorithms often perform better than unsupervised algorithm but at the expense of additional computational, memory, and time requirements. Nonetheless, some unsupervised training algorithms such as the GMM and the HMM are considered the state of the art classification techniques in the context of speaker recognition. In addition, unsupervised learning algorithms have an extra advantage over supervised ones in that new speakers can be added easily to the identification system without the need to retrain other speaker models.

In this section, different supervised and unsupervised classification techniques are reviewed. For unsupervised techniques, k-NN, the VQ, the HMM classifiers are briefly reviewed. The GMM is studied in details in the next chapter. For supervised techniques, the MLP, the RBF, and the SVM are studied as representatives for supervised learning algorithms that employ a single model for all classes.

2.3.1 Unsupervised learning techniques

As mentioned earlier, a basic advantage of unsupervised learning algorithms is in the flexibility of adding and removing speakers from the system. Unsupervised learning algorithms are often characterized by the decision function that is used to measure the match between a given testing utterance and a certain speaker model. While distance metrics are utilized with the NN, the dynamic time warping (DTW), and the VO classification methods, a probabilistic likelihood is used with statistical classifiers such as the GMM and the HMM.

Nearest Neighbor

The NN classification method is a conceptually simple classification technique that is found to be efficient in many pattern classification problems [33]. The training

phase just consists of storing all the training data vectors with their corresponding labels. To classify a testing example, the *closest* k training examples are found and a decision is made to the class that is most common in those k neighbors. Since a speech utterance is represented by a set of feature vectors, a distance metric between two sets of feature vectors should be defined. Given two sequences of feature vectors $U = \{\mathbf{u}[1], \ldots, \mathbf{u}[N_u]\}$ and $R = \{\mathbf{r}[1], \ldots, \mathbf{r}[N_r]\}$, Higgins defined the following metric for the task of speaker identification [58].

$$\begin{aligned} d(U,R) = & \frac{1}{N_u} \sum_{\mathbf{u}[i] \in U} \min_{\mathbf{r}[j] \in R} ||\mathbf{u}[i] - \mathbf{r}[j]||^2 + \frac{1}{N_r} \sum_{\mathbf{r}[j] \in R} \min_{\mathbf{u}[i] \in U} ||\mathbf{u}[i] - \mathbf{r}[j]||^2 \\ & - \frac{1}{N_u} \sum_{\mathbf{u}[i] \in R} \min_{\mathbf{u}[j] \in U, j \neq i} ||\mathbf{u}[i] - \mathbf{u}[j]||^2 - \frac{1}{N_r} \sum_{\mathbf{r}[i] \in R} \min_{\mathbf{r}[j] \in U, j \neq i} ||\mathbf{r}[i] - \mathbf{r}[j]||^2 \end{aligned} \quad (2.5)$$

The NN classification technique was applied to the KING and the Switchboard databases in [58]. The number of speakers in the KING database was 51 while 24 speakers (12 male and 12 female) were selected from the Switchboard database. For the KING database, the classification accuracy was 79.9% when the recoding equipments used with the training and testing utterances are the same and 68.1% when they are different. For the Switchboard, the recognition accuracy was 95.9%.

Since the k-NN classifier requires the storage of all the training data vectors, it is considered very costly in terms of the computational and memory requirements. Therefore, its implementation may be infeasible practical applications.

Vector quantization

In order to reduce the huge storage requirements inherent in the k-NN classification techniques, the training data may be divided into homogenous groups of each which is called a *cluster*. The center of each cluster, also called *centroid*, is then used to represent all the data vectors in this cluster. This way of *compression* is usually called vector quantization (VQ) in the context of speech recognition. This collection of centroids is called the codebook, which is a compact representation of the training data. The model of each speaker model just contains the codebook constructed

from its corresponding training data. There are many algorithms proposed for the codebook design such as the Linde-Buzo-Gray (LBG-VQ) method [72], the learning vector quantization (LVQ) method [69], and the group vector quantization (GVQ) [53]. The LBG-VQ method was applied in [130] to utterances of 35 speakers in the CSLU (center for spoken language understanding) database. There were mismatches between the speech utterances taken from different speakers and also between different recording sessions of the same speaker. A codebook of size 64 MFCC vectors was designed for each speaker. The obtained classification accuracy was 62.9%.

The VQ-methods do not consider the temporal profile of neither the training nor the testing utterances. Though this greatly simplifies the implementation of the identification, the temporal information is useful in speaker identification tasks [17]. This may be the reason for the relatively low accuracies obtained by the VQ methods.

Hidden markov model (HMM)

The HMM classifier has been extensively used in speech applications such as isolated word recognition and speech segmentation because it is physically related to the production mechanism of the speech signal [91]. Moreover, the temporal dynamics of the data are captured though state transitions.

The HMM is a doubly stochastic process which is compromised of a probabilistic finite state machine (Markov chain) in which each state is associated with another random variables producing the *observations*. Therefore, the main difference between the Markov chain and the HMM is in that the states are not directly observable and the observations are probabilistic functions in the state sequence. Usually, the observation random variable is either discrete or follow the GMM distribution [91, 88]. In discrete HMMs, the VQ codebook is first obtained from the training data. Each vector in the codebook is assigned a unique label. The set of the codebook labels forms the sample space of the observation random variables.

The set of parameters for discrete HMMs contains the initial state probabilities, the state transition probabilities, and the observation probabilities. For the continuous HMMs, the observation probabilities are replaced by the prior probabilities, the means vectors, and the covariance matrices of the observation GMM density. Training a speaker HMM is equivalent to finding the HMM parameters that maximizes the probability of the observations. The Baum-Welch estimation technique is the most widely used method for this task [91, 29]. In the recognition phase, the match function between a sequence of a testing feature vectors and a certain speaker is defined as the probability this sequence is generated by the corresponding speaker model. It should be mentioned that many variants of the HMM have been proposed and applied in voice recognition and other applications. In addition, very efficient algorithms have been developed for training HMMs and for calculating the likelihoods for sequence of data vectors (For a survey on HMMs, see [34] and the references therein).

The use of HMM in speaker recognition dates back to the eighties. In [89], an ergodic 5-state HMM (i.e., all possible transitions between states are allowed) was proposed by Poritz for this task. Tishby [111] expanded Poritzs idea by using an 8-state ergodic autoregressive HMM represented by continuous probability density functions with 2 to 8 mixture components per state. Matsui and Furui conducted a conducted a comparison between the VQ method, the discrete HMM, and the continuous HMM in terms of the classification accuracy and the robustness against noise. They found that the continuous HMM is far superior to the discrete HMM and as robust as the VQ-method. They also studied the effect of the number of mixtures and the number of Gaussian components per state on the identification accuracy. Upon their investigation, they concluded that the recognition accuracy is highly dependent on the number of Gaussian components but almost uncorrelated with the number of states. Therefore, they ended up with a conclusion that there is no significant difference in performance between the HMM and the GMM, which is an HMM with only one state. The robust classification performance of the GMM classifier was also reported by Rose et al. [96].

2.3.2 Supervised learning techniques

The main idea of supervised learning approaches is to learn the decision boundaries rather than the distribution of individual classes. Many supervised training algorithms are capable of generating a model that can distinguish one of M classes. Alternatively, a model can be generated for each speaker in the population so that it can distinguish between vectors in that class and vectors in all other classes. It has been found experimentally that the latter approach provides a higher classification performance [94]. Several supervised training algorithms have been investigated for speaker identification such as the MLP [108, 36], RBF [75, 127], and SVM [121, 128]. For the closed-set speaker identification problem, the performance obtained with the supervised training algorithms was typically comparable to the unsupervised techniques. However, the extensive training time necessary for most supervised algorithms is an undesirable feature. For tasks that require rejection capabilities, such as speaker verification and open set speaker identification, it was found that supervised methods consistently outperform the more traditional unsupervised methods [37, 36].

Multi-layer perceptron (MLP)

The MLP is a popular form of neural network that has been considered for various speech processing tasks [80, 73]. The structure of a MLP is illustrated in Fig. 2.5. The weights for MLPs are trained with the backpropagation algorithm [13] such that they can associate a high output response with particular input patterns.

For speaker recognition, the configuration of one-model-pre-class, described in the introductory paragraph in subsection 2.3.2, is usually employed with the MLP classifier. Ideally, the MLP for each speaker should output a one-response for the test feature vectors of that speaker and a zero-response for test vectors of other speakers. In the recognition phase, all test vectors are applied to each MLP and the outputs of each are accumulated. The speaker is selected as corresponding to the MLP with the maximum accumulated output. The use of the MLP classifier

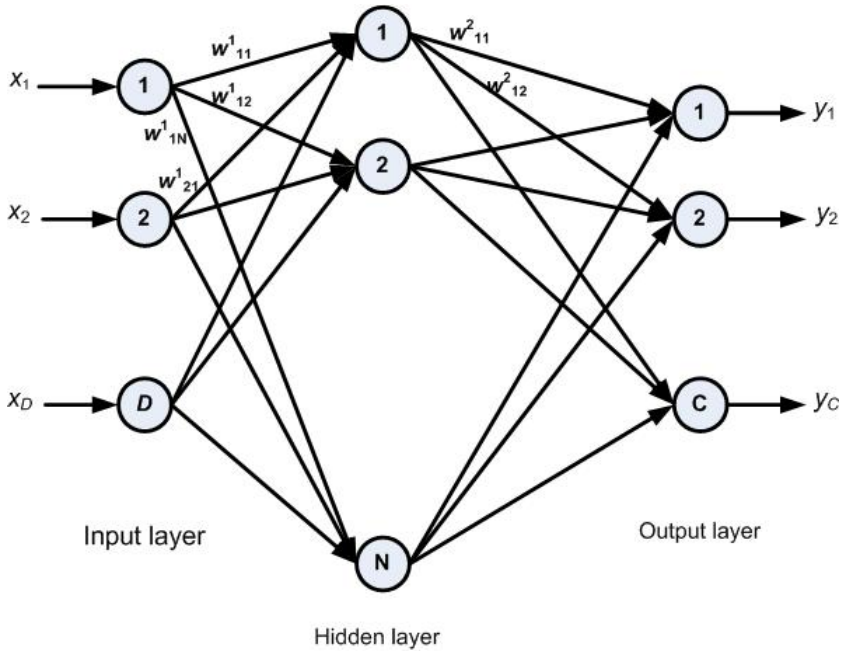

Figure 2.5: Multi-layer perceptron neural network.

for speaker identification problems was suggested in [84]. The speech for their simulations is drawn from a 10 speaker database and consists of 500 utterances from the digit set, 100 of which were used for training and 400 for recognition. A MLP with one hidden layer and 128 hidden nodes achieved a 92% identification rate for this experiment, which was just slightly worse than the performance obtained with a VQ classifier with 64 codebook entries per speaker. The performance improved as the number of hidden nodes increased. However, it was observed that increasing the number of hidden layers did not improve generalization. It was also noted that the performance of MLPs degrades rapidly as the speaker population increases.

Radial basis function (RBF)

Another major category of neural networks is the RBF networks. Basically, the RBF consists of three layers. The first layer is responsible for coupling the input vector to the network and has a linear neuron function. The last layer has a number of neurons equivalent to the number of speakers and uses an adjustable sigmoid as neuron function. In the hidden layer, a special function, called the RBF, is used

as an activation function. The RBF monotonically decreases with the increase of the distance to some specified centers, which are usually obtained by the k-means algorithm. For the proper choice of kernel function and perceptron weights, the RBF network becomes equivalent to the GMM with the exception that supervision is available here [13].

The RBF classifier was implemented in [96] and applied to subset of 16 speakers in the KING database. Using an RBF with 800 Gaussian basis functions, the average classification accuracy was 87.2%.

Support vector machines

The SVM is a statistical binary classifier that is based on the structural risk minimization (SRM) induction principle [119], which aims at minimizing a bound on the generalization error, rather than minimizing the training error. The SMV makes its decisions by constructing an optimal hyperplane that separates the two classes with the largest margin. In most classification problems, it is very difficult to find a separating hyper-plane in the original feature space. Therefore, a nonlinear mapping for the features to a higher dimensional space is usually performed before looking for the separating hyper-planes.

Recently, the SVM classifier has drawn much interest in many classification problems [33]. In text-independent speaker identification, the GMM has been a popular choice for the nonlinear kernel mapping function [128, 39]. However, other functions such as the linear kernel, polynomial and radial basis kernel are also used [121].

Another important issue is that the theory of the SVM classifier was mainly developed for the binary classification problem [119, 33]. Basically, there are two main approaches for generalization to the multi-class SVM classification system. In the first approach, each possible pair of the classes is used to train a SVM classifier. That is, if the total number of speakers is S, the total number of the binary SVM models is $S(S-1)/2$. For a test utterance, the pairwise comparison [70]

strategy is adopted to identify its speaker. Clearly, when the number of speakers is relatively large, the computational requirements of both the training and the testing algorithms of this approach become excessive. Unfortunately, this is typically the case in most speaker identification problems.

Alternatively, one can employ the method described in the first paragraph in 2.3.2. In this case, a bit inferior classification performance should be expected. The SVM was applied in [121] to utterances of twenty speakers (10 males and 10 females) selected from the AURORA-2 database. The radial basis kernel functions were adopted in the experiment. The classification accuracy was 90.1% for clean speech and 48.6% for artificially corrupted speech (after enhancing the speech quality).

2.4 Mismatch reduction techniques

During the last two decades, there has been extensive research on reducing the effect of handset channel mismatch, which significantly hamper the performance of speaker recognition systems. In general, compensation techniques can be grouped into three categories: feature-based, model-based, and score-based compensation techniques. In this section, we give a brief review about methods in each category. It should be mentioned that compensation techniques are not exclusive in general. That is, it is possible to combine techniques that belong to two or more different domains, e.g. feature-based and model-based, so as to achieve an even better compensation [100].

2.4.1 Feature-based compensation techniques

In feature-based compensation, the goal is to derive features that are insensitive as possible to non-speaker related factors such as the handset type, sentence content, and channel effects. At the same time, they should provide good discrimination between different speakers.

In this brief review, we shall cover only three of the most standard (and classical) feature-based compensation techniques: the SS method [14], the cpestral mean normalization (CMS) method [41] and the RASTA-PLP method [57]. Other feature-based compensation methods include discriminative feature design [54], feature warping [86], and short-time Gaussianization [123].

The SS technique assumes that the noise is stationary and it affects the energy contour of the noisy signal in an additive way. Hence the additive noise component could be subtracted from the noisy speech energy to estimate the clean speech energy. The additive noise component is generally computed from the silence portion of the speech. In reality, the stationary assumption does not hold. Hence, it is possible that the noise energy in some frequency bins can exceed that of the noisy speech resulting in a negative estimate of the clean speech energy. This necessitates the use of a floor value. The floor value is expressed as a portion of the noise energy. Let $|S(e^{j\omega})|^2$, $|N(e^{j\omega})|^2$, and $|X(e^{j\omega})|^2$ be the energies of the clean speech, the noise, and the noisy speech, respectively. According to the SS method, an estimate for the energy of the clean signal is given by

$$|\hat{S}(e^{j\omega})|^2 = \max\left\{|X(e^{j\omega})|^2 - |N(e^{j\omega})|^2, k|N(e^{j\omega})|^2\right\}, \tag{2.6}$$

where k is an empirical constant, which is usually less than one [85]. It has been found that performance of the SS method heavily depends on the floor value, $k|N(e^{j\omega})|^2$ [31]. Therefore, statistical methods have been proposed for the estimation of the noise floor [32, 125].

The CMN method depends on the fact that the filtering effect of the communication channel is equivalent to an additive vector in the mel-cepstral domain [103]. Thus, the channel effect can be removed by subtracting the mean cepstral vector from all the cepstral feature vectors extracted from each utterance. As a consequence, all feature vectors have the same mean vector and performance is not affected by the channel biases. When additive noise exists, a natural extension to the CMN is the cepstral mean and variance normalization (CMVN) [107], which normalizes the distribution of cepstral features over some specific window length

by subtracting the mean and dividing by the standard deviation.

RASTA is a modulation spectrum analysis that aims to reduce the effects of convolutional noise in the communication channel. This achieved by 1) attenuating low modulation frequency components and 2) enhancing the dynamic parts of the spectrogram [56]. Similar to the CMN, the low frequency components are filtered out in order to remove the additive channel-dependent vector. It has been claimed that the second property is also beneficial for good recognition performance [56]. The classical RASTA filter has the following transfer function [56].

$$H(z) = 0.1z^4 \frac{2 + z - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \tag{2.7}$$

This transfer function introduces phase distortion, which causes time masking for the auditory human perception. Therefore, a phase-correction step was suggested in [26] after the RASTA calculation. The use of both the CMN and the RASTA processing methods has been much recommended in many speaker recognition systems [96, 100].

2.4.2 Model-based compensation techniques

Model-based compensation techniques attempt to reduce the effect of channel variations by learning channel characteristics or enhancing the speaker probability distribution models. The most two well known examples in this category are speaker model synthesis (SMS) [110] and PMC [43].

The SMS technique learns how the speaker model parameters change among different channels, and uses this information to synthesize speaker models for channels where no enrollment data is available. It utilizes channel-dependent UBMs as a priori knowledge of channels for speaker model synthesis. This algorithm assumes that all the speakers are subject to the same model transformation between two different channels; however in reality different speakers may be subject to different model transformations.

The PMC approach attempts to estimate the corrupted speech model by combining the clean speech model with a background noise model. The PMC is much related to the extraction process of the MFCC features. Therefore, the following domains are defined: the linear-spectral domain, the log-spectral domain, and the cepstral domain (see figure 2.4). A diagram showing the basic process of the PMC is shown in figure 2.6. The inputs to the scheme are the clean speech models and the noise model. Usually, the combination of speech and noise is expressed in either the linear-spectral domain or the log-spectral domain. Hence, the combination of the noise and clean parameters are made in one of these two domains. After combination of parameters, the estimates of the corrupted speech parameters are transformed back to the cepstral domain if required. The PMC has been shown to achieve good performance in speech recognition and speaker recognition applications [83, 44]. However, a drawback of the PMC is that it assumes the availability of an accurate statistical model for the noise in the training phase. This assumption is not valid for many practical applications since the training and the testing utterances of the same speaker may well be recorded in different environments.

2.4.3 Score-based compensation techniques

While feature-based compensation techniques address linear channel effects, the handset transducer effects are nonlinear in nature and are thus difficult to remove from features before training and recognition [90]. As a result, the speaker's model represents the speaker's acoustic characteristics coupled with the distortions caused by the handset from which the training speech was collected. This coupling introduces handset-dependent biases and scales to the likelihood scores of the speaker acoustics models. Therefore, score-domain compensation aims to remove handset-dependent biases from the likelihood scores. The most prevalent methods in this category include the H-norm method [100], the Z-norm method [9], and the T-norm method [5].

The H-norm score normalization technique works as follows. All the speakers are

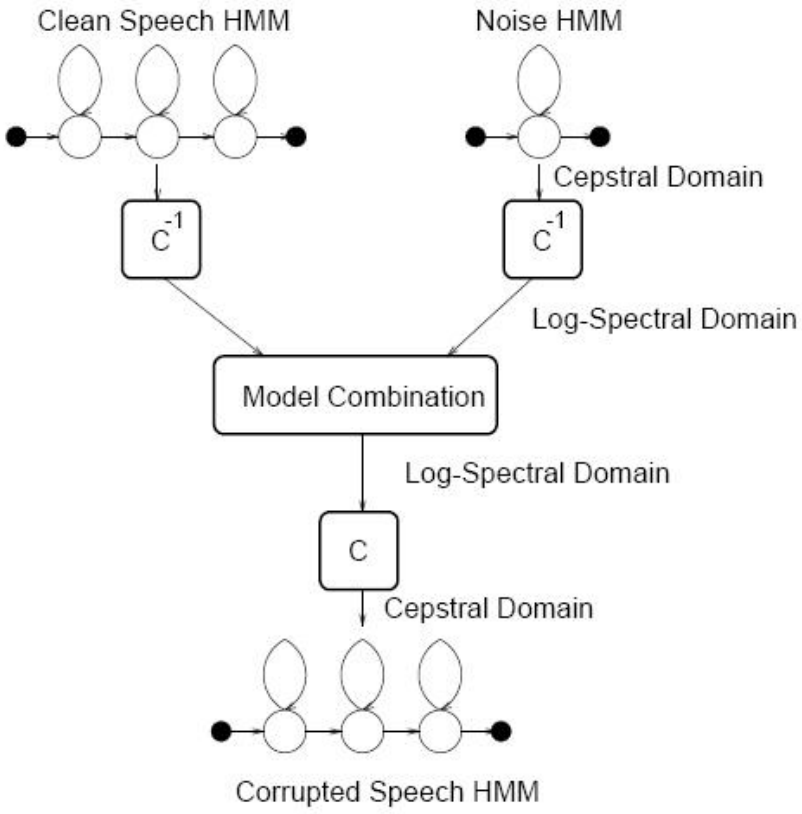

Figure 2.6: The basic parallel model combination process.

grouped according to the handset they used for producing the training utterances. For each handset type, the scores of all speakers in the corresponding group are calculated and then normalized according to the following relation

$$s_{\text{H-Norm}}(X) = \frac{\log P(X|\lambda) - \mu_h}{\sigma_h}, \tag{2.8}$$

where μ_h and σ_h are the average and the standard deviation of the scores in this group. The Z-norm and T-norm are given by relations similar to (2.8). The only difference is in the definition of the normalizing factors. Both the Z-norm and the T-norm techniques use a set of cohort speakers who are close to the target speaker. The selection of the cohort can be done during training when the speaker model is compared to cohort models using a similarity measure [5]. In the Z-norm technique, the scores are defined as the log-likelihood of the target speaker model with respect to the utterances of the cohort speakers. Meanwhile, in the T-norm technique, the

scores are defined as the log-likelihood of the cohort speakers' models with respect to the testing utterance.

2.5 Summary and conclusions

In this chapter, we presented a brief review about closed-set text-independent speaker identification. We explained the MFCC feature extraction process because it is one of the most popular features used for speaker recognition. In addition, we surveyed the common classification techniques used in the context of speaker identification as well as the different types of mismatch reduction techniques. From this survey, we conclude that the classification performance of real world speaker identification systems still needs much improvement. Moreover, despite the relative improvement in mismatch reduction achieved by feature-based and channel-based methods, it seems that they do not provide much space for further progress. Therefore, we main attention in this thesis was toward model-based compensation.

Chapter 3

Gaussian Mixture models

Gaussian Mixture Model (GMM) has become a dominant approach for speaker recognition problems [103]. Several reasons are attributed to this dominance. Among them are the achieved robustness and the possibility to model the underlying acoustic classes. Moreover, a well-established mathematical basis has been developed for GMMs. In general, two main frameworks have been proposed for GMM-based speaker identification: the standard ML decision framework and the Gaussian mixture model/universal background model (GMM/UBM) framework.

This chapter gives an overview of both classification frameworks. Section 3.1 defines analytically the GMM. The standard ML and the GMM/UBM frameworks are described in details in section 3.2 and section 3.3, respectively.

3.1 Mathematical definition of the GMM

A mixture model of order M is a convex combination of M probability density functions in the form:

$$\mathrm{p}(\mathbf{x}|\lambda) = \sum_{m=1}^{M} w_m \mathrm{p}(\mathbf{x}|m, \lambda) \tag{3.1}$$

where $\mathbf{x}$ is a D-dimensional vector, λ is a string representing the model parameters, $\mathrm{p}(\mathbf{x}|\lambda)$ is the model density function, $\mathrm{p}(\mathbf{x}|m, \lambda)$ is the density function of the m^{th}

component, and w_m is the *a priori* probability of the m^{th} Gaussian component density, or simply, the weight of the i^{th} component. The prior probabilities must be nonnegative and sum to one so that (3.1) is a valid probability density function. For the case of GMM, $\mathrm{p}(\mathbf{x}|m,\lambda)$ is the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$,

$$\begin{aligned}\mathrm{p}(\mathbf{x}|m,\lambda) &\equiv \mathbb{N}(\mathbf{x};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)\\ &= \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_m|^{1/2}}\exp\left((\mathbf{x}-\boldsymbol{\mu}_m)^{\mathsf{T}}\boldsymbol{\Sigma}_m^{-1}(\mathbf{x}-\boldsymbol{\mu}_m)\right). \qquad (3.2)\end{aligned}$$

Thus, a GMM with M mixtures is parameterized by a set of M positive weights that sum to unity, M mean vectors, and M covariance matrices. These parameters are collectively represented by the string

$$\lambda = \{w_1,\ldots,w_M,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_M,\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_M\}. \qquad (3.3)$$

There are three types of GMMs depending on the choice of covariance matrices. The model can have one covariance matrix per Gaussian component (nodal covariance), one covariance matrix for all Gaussian components in a speaker model (grand covariance), or one covariance matrix shared by all speaker models (global covariance). The covariance matrix can also be full or diagonal. GMMs with nodal covariance matrices are primarily used in our study.

3.2 Standard maximum likelihood framework

As mentioned in the introduction, the speaker identification problem can be formulated as a multi-hypothesis classification problem. In the standard ML framework, each speaker (hypothesis) is modeled by a GMM. In the training phase, the feature vectors of each speaker are used to estimate his/her model parameters based on the ML estimation principle. In the testing phase, the ML decision rule is used to identify the speaker of the testing utterance. In this section, we address the parameter estimation problem and classification using the ML-decision rule.

3.2.1 Parameter estimation

Let $X = \{\mathbf{x}[1 : N]\}$ denote the set of the training feature vectors of a certain speaker[1]. In GMM-based speaker identification systems, it is assumed that all the feature vectors are statistically independent, i.e.,

$$\begin{aligned} \mathrm{p}(\mathbf{x}[1:N]|\lambda) &= \prod_{n=1}^{N} \mathrm{p}(\mathbf{x}[n]|\lambda) \\ &= \prod_{n=1}^{N} \left(\sum_{m=1}^{M} w_m \mathbb{N}\left(\mathbf{x}[n]; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right) \right). \end{aligned} \tag{3.4}$$

Obviously, the above likelihood function is a highly nonlinear function in the model parameters. Hence, maximization of the likelihood function is only possible through iterative procedures such as gradient-based methods and the EM algorithm.

The EM algorithm, proposed by Dempster *et al.* [29], basically depends on the existence of 'complete' data set Z from which the given data X can be derived. For the problem in hand, the complete data specification is $Z = \{\Phi, X\}$, where $\Phi = \{\phi[1 : N]\}$, and $\phi[n]$ is the index of the Gaussian component selected at time n. The basic idea of the EM algorithm is to start with some initial model $\lambda^{(0)}$ and look for another model $\lambda^{(1)}$ with a higher likelihood value. Dempster *et al.* proved that for any model $\lambda^{(s)}$, the model $\lambda^{(s+1)}$ obtained by maximizing the following auxiliary function must have an equal or larger likelihood function.

$$Q(\lambda; X, \lambda^{(s)}) = E\left\{\log P\left(X, \Phi|\lambda\right) | X, \lambda^{(s)}\right\}. \tag{3.5}$$

This is one iteration of the algorithm. Starting from an initial model $\lambda^{(0)}$, the auxiliary function $Q(\lambda; X, \lambda^{(0)})$ is constructed and then optimized with respect to λ. The obtained model $\lambda^{(1)}$ will be the initial model for the next iteration in which another auxiliary function $Q(\lambda; X, \lambda^{(1)})$ is constructed and optimized again with respect to λ and so on. Since this iterative procedure always guarantees an increase in the incomplete likelihood function thanks to Dempster theory, the EM should stop when a maximum number of iterations is exceeded or the increase in the likelihood function is less than a small threshold.

[1]For convenience, we dropped the speaker index in this section.

The EM update equations are derived as follows. The complete likelihood function is given by

$$\begin{aligned} P\left(X, \Phi|\lambda\right) &= \prod_{n=1}^{N} P\left(\mathbf{x}[n], \phi[n]|\lambda\right) \\ &= \prod_{n=1}^{N} P(\phi[n]|\lambda) P(\mathbf{x}[n]|\phi[n], \lambda) \\ &= \prod_{n=1}^{N} w_{\phi[n]} \mathbb{N}(\mathbf{x}[n]; \boldsymbol{\mu}_{\phi[n]}, \boldsymbol{\Sigma}_{\phi[n]}). \end{aligned} \tag{3.6}$$

Substituting (3.6) into (3.5), and simplifying, the auxiliary function is given by

$$\begin{aligned} &Q(\lambda; X, \lambda^{(s)}) = \\ &c + \sum_{m,n} P_{m,n}(\lambda^{(s)}) \left(\log w_m - \frac{1}{2} \log |\boldsymbol{\Sigma}_m| - \frac{1}{2} (\mathbf{x}[n] - \boldsymbol{\mu}_m)^{\mathrm{T}} \boldsymbol{\Sigma}_m^{-1} (\mathbf{x}[n] - \boldsymbol{\mu}_m) \right), \end{aligned} \tag{3.7}$$

where $\sum_{m,n}$ is a shorthand for $\sum_{m=1}^{M} \sum_{n=1}^{N}$, c is an irrelevant constant, $P_{m,n}(\lambda^{(s)})$ is the a posteriori probability of the m^{th} Gaussian component given the observation $\mathbf{x}[n]$, i.e.,

$$\begin{aligned} P_{n,m}(\lambda) &\equiv P\left(\phi[n] = m \,|X; \lambda\right) \\ &= \frac{w_m \mathbb{N}(\mathbf{x}[n]; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{m'=1}^{M} w_{m'} \mathbb{N}(\mathbf{x}[n]; \boldsymbol{\mu}_{m'}, \boldsymbol{\Sigma}_{m'})}. \end{aligned} \tag{3.8}$$

The EM update equations are obtained by maximizing the auxiliary function in (3.7) with respect to λ. Fortunately, the auxiliary function is uni-modal in λ, and hence $\lambda^{(s+1)}$ is obtained simply by differentiating $Q(\lambda; X, \lambda^{(s)})$ with respect to λ and equating to zero. Regarding the model priors, however, there is an additional constraint, which is,

$$\sum_{m=1}^{M} w_m^{(s+1)} = 1.$$

Hence, the update equations for the priors are obtained by maximizing the following Lagrangian function

$$\mathrm{Q}'(\lambda; X, \lambda^{(s)}) = \mathrm{Q}(\lambda; X, \lambda^{(s)}) + \beta \left(\sum_{m=1}^{M} w_m - 1 \right).$$

It is straightforward to show that, upon differentiating $\mathrm{Q}'(\lambda; X, \lambda^{(s)})$ with respect to $w_m, m = 1, \ldots, M$, the update equations for the model priors are given by

$$w_m^{(s+1)} = \frac{1}{N} \sum_{m,n} P_{n,m}(\lambda^{(s)}), \qquad m = 1, 2, ..., M. \tag{3.9}$$

Similarly, the update equations for the model centers and covariance matrices are given by

$$\boldsymbol{\mu}_m^{(s+1)} = \frac{\sum_{n=1}^{N} P_{n,m}(\lambda^{(s)}) \mathbf{x}[n]}{\sum_{n=1}^{N} P_{n,m}(\lambda^{(s)})} \tag{3.10}$$

$$\boldsymbol{\Sigma}_m^{(s+1)} = \frac{\sum_{n=1}^{N} P_{n,m}(\lambda^{(s)})(\mathbf{x}[n] - \boldsymbol{\mu}_m^{(s+1)})(\mathbf{x}[n] - \boldsymbol{\mu}_m^{(s+1)})^{\mathsf{T}}}{\sum_{n=1}^{N} P_{n,m}(\lambda^{(s)})}. \tag{3.11}$$

In (3.11), no assumption is made regarding the structure of the covariance matrices, $\boldsymbol{\Sigma}_m, m = 1, \ldots, M$. However, in order to reduce the computational requirement for both the training and the testing algorithms, the covariance matrices are usually assumed to be diagonal [96, 103]. It is also argued that the classification performance of GMM-based system with diagonal covariance matrices is superior to those with full covariance matrices [100]. Note that the diagonal assumption does not imply the statistical independence between the feature components (dimensions) since, at any time instant n, the index of the selected Gaussian component is the same for all dimensions. However, in order to represent the same distribution, the number of Gaussian components with diagonal covariance is much more than the number of Gaussian components with full covariance. Let the diagonal covariance matrix of the m^{th} Gaussian component be

$$\boldsymbol{\Sigma}_m = \mathrm{diag}(\sigma_{m,1}^2, \ldots, \sigma_{m,D}^2).$$

In this case, the normal distribution $\mathbb{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ simplifies to

$$\mathbb{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \prod_{d=1}^{D} \mathbb{N}(x_d; \mu_{m,d}; \sigma_{m,d}^2) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{m,d}^2}} \exp\left(-\frac{(x_d - \mu_{m,d})^2}{\sigma_{m,d}^2} \right). \tag{3.12}$$

Following the same steps used to derive (3.11), the update equations for $\sigma_{m,d}^2$ is given by

$$(\sigma_{m,d}^2)^{(s+1)} = \frac{\sum_{n=1}^{N} P_{n,m}(\lambda^{(s)})(x_d[n] - \mu_{m,d}^{(s+1)})^2}{\sum_{n=1}^{N} P_{n,m}(\lambda^{(s)})}. \tag{3.13}$$

It should be mentioned that the EM algorithm is a local optimization algorithm. That is, the ML estimate of the model parameters is sensitive to the initial model estimate $\lambda^{(0)}$. One method to alleviate this problem is to use the k-means algorithm [72] to divide the training data into M clusters. The initial estimates of the parameters of each Gaussian component are estimated from the corresponding cluster as follows [13]

$$w_m^{(0)} = N_m/N. \tag{3.14}$$

$$\boldsymbol{\mu}_m^{(0)} = \frac{1}{N_m} \sum_{n \in \mathcal{C}_m} \mathbf{x}[n], \tag{3.15}$$

$$\boldsymbol{\Sigma}_m^{(0)} = \frac{1}{N_m} \sum_{n \in \mathcal{C}_m} \mathbf{x}[n]\mathbf{x}^{\mathtt{T}}[n] - \boldsymbol{\mu}_m^{(0)}(\boldsymbol{\mu}_m^{(0)})^{\mathtt{T}}, \tag{3.16}$$

where C_m is the set of the indices of the data points in the m^{th} cluster and N_m is the number of data points in this cluster. For diagonal covariance matrices, the initial estimate of the diagonal entries is given by

$$(\sigma_{m,d}^{(0)})^2 = \frac{1}{N_m} \sum_{n \in \mathcal{C}_m} x_d^2[n] - \left(\mu_{m,d}^{(0)}\right)^2. \tag{3.17}$$

3.2.2 Classification framework

In the standard ML classification framework, the index of the decided speaker is determined according to the ML decision rule. Given a sequence of testing vectors, $X = \{\mathbf{x}[1:N]\}$, extracted from an unknown speech utterance the required speaker model should attain the largest *a posteriori* probability. Using Bayes' theorem, the index of the selected speaker is:

$$\hat{s} = \arg \max_{s=1,\ldots,S} P(\mathcal{H}_s|X) = \arg \max_{s=1,\ldots,S} \frac{\mathrm{p}(X|\mathcal{H}_s)\, P(\mathcal{H}_s)}{\mathrm{p}(X)}. \tag{3.18}$$

The denominator in the above equation is irrelevant to the maximization argument, s, and hence it can be dropped. In addition, in most applications there is no reason to favor a speaker over another *a priori*, and hence, the prior probability of all hypotheses should be the same. Thus, equation (3.18) reduces to

$$\hat{s} = \arg \max_{s=1,\ldots,S} \mathrm{p}(X|\mathcal{H}_s) = \arg \max_{s=1,\ldots,S} \prod_{n=1}^{N} \mathrm{p}(\mathbf{x}[n]|\mathcal{H}_s)\,. \tag{3.19}$$

In the above equation, a product of a large number of small values should be evaluated for each speaker. Therefore, direct implementation of the above decision rule on a digital computer results in an underflow. Alternatively, maximization may be made with respect to $\log \mathrm{p}(X|\mathcal{H}_s)$ yielding

$$\begin{aligned}\hat{s} &= \arg \max_{s=1,\ldots,S} \sum_{n=1}^{N} \log \mathrm{p}(\mathbf{x}[n]|\mathcal{H}_s) \\ &= \arg \max_{s=1,\ldots,S} \sum_{n=1}^{N} \log \left(\sum_{m=1}^{M} P(\phi[n]=m|\mathcal{H}_s) \mathrm{p}(\mathbf{x}[n]|\phi[n]=m,\mathcal{H}_s) \right),\end{aligned}$$

where $P(\phi[n]=m|\mathcal{H}_s)$ is the prior probability (weight) of the m^{th} Gaussian component in λ_s and $\mathrm{p}(\mathbf{x}[n]|\phi[n]=m,\mathcal{H}_s)$ is given by (3.2) for full covariance matrices and (3.12) for diagonal covariance matrices. Hence, we need to evaluate a summation in the form $\sum_{m=1}^{M} e^{-a_m}$ for large a_m. This can be done without encountering computer underflows by using the Jacobian log [35]. The basic idea is to add logs one at a time, as follows

$$\begin{aligned}a_{12} &= \log(e^{a_1} + e^{a_2}) \\ &= \max(a_1, a_2) + \log(1 + e^{-|a_1 - a_2|})\end{aligned}$$

Then, the new exponential a_3 is added to a_{12} the same way and so on. A diagram of the ML classification framework is illustrated in figure 3.1.

3.3 The Gaussian mixture model/universal background model framework

In the standard ML framework, it is implicitly assumed that no prior information about the model parameters is available. In some applications such as the considered speaker identification problem, incorporating prior information about the model parameters improves the classification performance. The maximum a posteriori (MAP) estimation principle is a direct generalization to the ML estimation

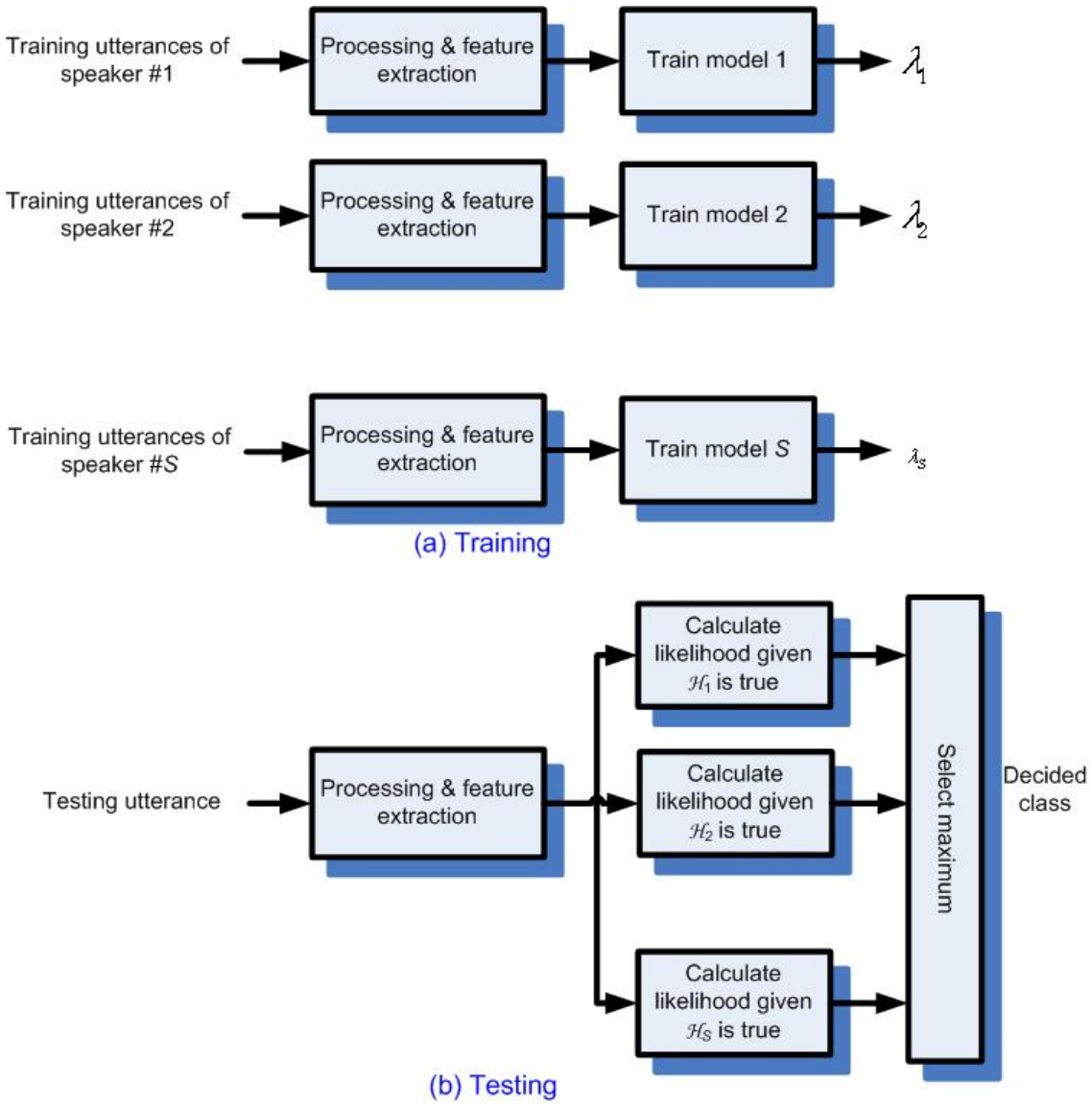

Figure 3.1: The standard ML framework for speaker identification. (a) Training sub-system. (b) Testing sub-system.

principle in which the model parameters are considered as random quantities with some given prior distribution.

In the context of speaker identification (and verification), the prior distribution of the parameters of all speaker models are assumed the same and derived from a speaker-independent distribution called the universal background model (UBM) [103]. There are many ways to construct the UBM. The simplest method is to pool the training data of all speakers together and use them to train a single

GMM. However, we should be careful of the distribution of the sub-populations in the training database. For example, if number of utterances of female speakers is much more than that of male speakers, the UBM constructed by the above method will be biased towards the female distribution. A similar issue applies to utterances with different microphones and different recording environments (if known). Some other approaches model each subpopulation separately. The UBM parameters are estimates as a convex combination of the parameters of the sub-populations models. The mixing weights should be carefully selected to reflect the proportion of each sub-population. In this thesis, we considered only utterances recorded using electret microphones. In addition, we performed our experiments using male-only database, female-only database, or mixed-gender database with almost equal proportions of males and females. Therefore, UBM is trained using the training data of all speakers.

Thus, the training phase in the GMM/UBM framework consists of two steps. In the first step, all the training data are combined together and used to train a GMM/UBM model,

$$\lambda_{\text{UBM}} = \{\tilde{w}_1, \ldots, \tilde{w}_M, \tilde{\boldsymbol{\mu}}_1, \ldots, \tilde{\mathbf{w}}_M, \tilde{\boldsymbol{\Sigma}}_1, \ldots, \tilde{\boldsymbol{\Sigma}}_M\} \tag{3.20}$$

using the ordinary EM algorithm, where the tilde in (3.20) refers to the parameters of the UBM. It should be mentioned that, for large databases such as the 2000 NIST speaker evaluation used in our first experiment, direct implementation of the update equations in section 3.1 may require excessive space requirements. Hence, special memory conservative algorithms should be developed. This is internally implemented in our simulations but we prefer to omit these details for convenience.

Based on the obtained GMM/UBM, prior distributions of the model parameters are derived as follows. The distribution of the model priors is usually assumed in the Dirichlet form [63].

$$P(w_1, \ldots, w_M | \nu_1, \ldots, \nu_M) \propto \prod_{m=1}^{M} w_m^{\nu_m - 1}, \tag{3.21}$$

where $\nu_m > 0$ are the parameters of the Dirichlet density. Meanwhile, the prior distribution of the centers and the covariance matrices of the speaker model is assumed to follow a Wishart distribution [27, 47].

$$P(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m|\tau_m, \alpha_m, \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Sigma}}_m) \propto$$
$$|\boldsymbol{\Sigma}_m|^{(\alpha_m - d - 1)/2} \exp\left(-\frac{\tau_m}{2}(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m)^{\mathsf{T}}(\tilde{\boldsymbol{\Sigma}}_m)^{-1}(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m)\right)$$
$$\exp\left(-\frac{1}{2}\text{trace}(\boldsymbol{\Sigma}_m^{-1}\tilde{\boldsymbol{\Sigma}}_m)\right). \tag{3.22}$$

Assuming statistical independence between the model priors and the model centers and covariance matrices, the joint density of all the GMM parameters is the product of (3.21) and (3.22), i.e.,

$$P(\lambda|\theta) = P(w_1, \ldots, w_M|\nu_1, \ldots, \nu_M) \prod_{m=1}^{M} P(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m|\tau_m, \alpha_m, \tilde{\boldsymbol{\mu}}_m, \alpha_m, \tilde{\boldsymbol{\Sigma}}_m), \tag{3.23}$$

where $\theta = \{\nu_1, \ldots, \nu_M, \alpha_1, \ldots, \alpha_M, \tau_1, \ldots, \tau_M\}$.

In the second step of the training phase, the parameters of each speaker model are obtained by adapting the UBM using the speaker training data. Given a sequence of training feature vectors of a certain speaker, $X = \{\mathbf{x}[1:N]\}$, a generalized version of the EM algorithm is used to adapt the UBM. In particular, the distribution of the model parameters is included in the EM auxiliary function. The generalized auxiliary function in this case is given by [47]

$$Q(\lambda; X, \lambda^{(s)}, \theta) = E\left\{\log P\left(X, \Phi|\lambda, \theta\right) + \log P\left(\lambda|\theta\right)\Big|X, \lambda^{(s)}, \theta\right\}. \tag{3.24}$$

Substituting (3.23) into (3.24), it is not hard to show that

$$Q(\lambda; X, \lambda^{(s)}, \theta) =$$
$$c + \sum_{m,n} P_{m,n}(\lambda^{(s)}) \left(\log w_m - \frac{1}{2}\log|\boldsymbol{\Sigma}_m| - \frac{1}{2}(\mathbf{x}[n] - \boldsymbol{\mu}_m)^{\mathsf{T}}\boldsymbol{\Sigma}_m^{-1}(\mathbf{x}[n] - \boldsymbol{\mu}_m)\right)$$
$$+ \sum_{m=1}^{M}\left[(\nu_m - 1)\log w_m + \frac{\alpha_m - d - 1}{2}\log|\boldsymbol{\Sigma}_m| - \frac{\tau_m}{2}(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m)^{\mathsf{T}}\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m)\right.$$
$$\left. - \frac{1}{2}\text{trace}\left(\boldsymbol{\Sigma}_m^{-1}\tilde{\boldsymbol{\Sigma}}_m\right)\right] \tag{3.25}$$

Similar to the standard ML estimation, the GMM parameters are obtained by simply differentiating the above auxiliary function with respect to the different parameters and equating to zero. The constraint that the all the priors are positive and sum to unity still applies. The following update equations are easily obtained.

$$w_m^{(s+1)} = \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \nu_m - 1}{N - M + \sum_{m=1}^{M} \nu_m} \tag{3.26}$$

$$\boldsymbol{\mu}_m^{(s+1)} = \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)})\mathbf{x}[n] + \tau_m \tilde{\boldsymbol{\mu}}_m}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \tau_m} \tag{3.27}$$

$$\begin{aligned}\boldsymbol{\Sigma}_m^{(s+1)} = & \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)})\mathbf{x}[n]\mathbf{x}^{\mathtt{T}}[n] + \tau_m(\tilde{\boldsymbol{\mu}}_m\tilde{\boldsymbol{\mu}}_m^{\mathtt{T}} + \tilde{\boldsymbol{\Sigma}}_m)}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \alpha_m - d - 1} \\ & - \frac{\left(\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \tau_m\right)\boldsymbol{\mu}_m^{(s+1)}(\boldsymbol{\mu}_m^{(s+1)})^{\mathtt{T}}}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \alpha_m - d - 1}.\end{aligned} \tag{3.28}$$

In the last equation, full covariance matrices are considered. For diagonal covariance matrices, the update equation is

$$\sigma_d^{2(s+1)} = \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)})x_d^2[n] + \tau_m(\mu_{m,d}^2 + \tilde{\sigma}_{m,d}^2) - \left(\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \tau_m\right)(\mu_{m,d}^{(s+1)})^2}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \alpha_m - d - 1}. \tag{3.29}$$

The classification phase in the GMM/UBM framework is identical to that of the ML framework. A diagram showing the architecture of the GMM/UBM framework is shown in figure 3.2.

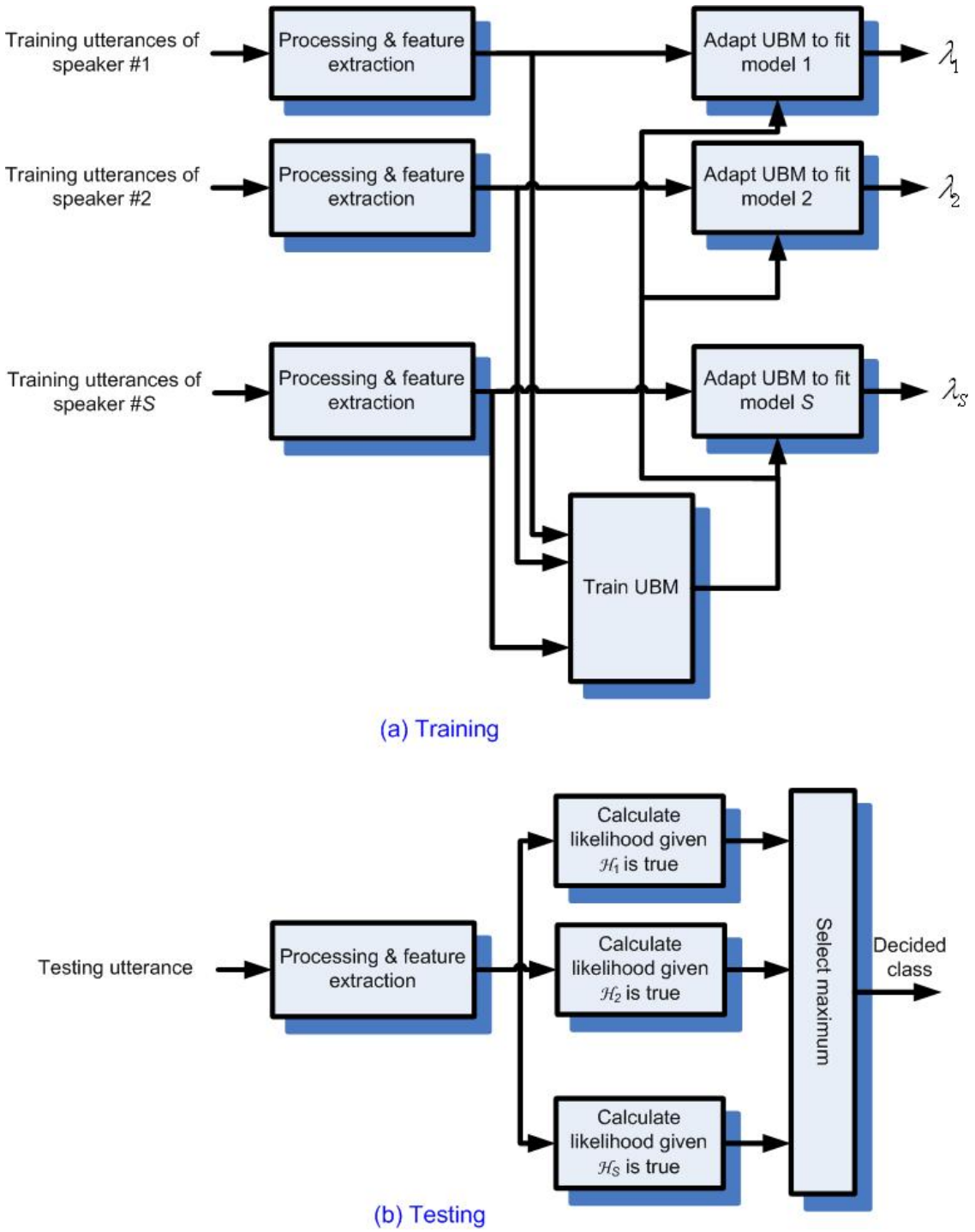

Figure 3.2: The GMM/UBM framework for speaker identification. (a) Training sub-system. (b) Testing sub-system.

Chapter 4

Vector autoregressive Gaussian Mixture model

In this chapter, we give a thorough analysis of the proposed vector autoregressive Gaussian mixture (VARGM) model and generalize the classification frameworks described in chapter 2 to handle VARGM models instead of GMM. The proposed VARGM model is a generalization to the VAR model in that the distribution of the innovation sequence, also called residual vectors, is a GMM instead of the multivariate normal distribution.

VAR model is a classical and simple tool, successfully used in characterizing and analyzing stationary multivariate time series data. It has been utilized in many applications such as signal processing [106], digital communication [114], and time series prediction [30]. The main reasons for their popularity are their simple structure and the availability of well established parameter estimation algorithms such as the maximum likelihood estimation (MLE) procedure and the Yule-Walker algorithm. The basic idea of the VAR model is to model the time series as the output of an all-pole linear time invariant filter whose input is a white Gaussian noise. This input is usually called the *innovation sequence*. That is, it is assumed that the individual samples of the innovation sequence are statistically independent and follow the multivariate Gaussian distribution.

In many speech applications, VAR models have been extensively employed to characterize the correlation between successive speech feature vectors. In fact, autoregressive Markov modeling of speech was originally proposed by Poritz [89]. His model consists of a sequence of states of each which is modelled by a VAR model rather than a GMM. Various modifications have been proposed to this model such as non-stationary autoregressive hidden Markov model (NAR-HMM) [71] and autoregressive hidden Markov model with duration [35].

However, the assumption of a white-Gaussian innovation sequence seems to be restrictive for speaker identification. Therefore, in this chapter, we relax this assumption by allowing the distribution of the innovation sequence to be in the form of a mixture of multivariate Gaussian densities. In fact, this model is also a vector generalization for the model proposed in [120]. Another advantage of the VARGM model is its ability to resemble a wide range of non-Gaussian VAR models. This is based on the fact that many distributions can be well approximated by a convex combination of Gaussian densities under some mild conditions [96].

The chapter is organized as follows. Section 4.1 briefly reviews VAR models and the estimation of model parameters based on the MSE criterion. In section 4.2, parameter estimation of the VARGM model parameters using the EM algorithm is explained. In section 4.3, we present a novel procedure for model order selection. Classification using the proposed VARGM classifier is discussed in section 4.4.

4.1 Vector autoregressive models

A vector time series $\mathbf{x}[1:N]$, $\mathbf{x}[n] \in \mathbb{R}^D$, i.e. D is the dimensionality of the vector $\mathbf{x}[n]$, can be modelled by a VAR model of order P of the following form [95].

$$\mathbf{x}[n] = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{x}[n-p] + \mathbf{e}[n] + \boldsymbol{\delta}, \qquad n = 1, \ldots, N, \tag{4.1}$$

where the vector $\boldsymbol{\delta}$ is called the *intercept vector* and P is the regression order. The vectors $\mathbf{e}[1:N]$ are called the *residual vectors*. Usually, the residual vectors are

assumed to be drawn from a white Gaussian process, i.e., they satisfy the following relations

$$E\{\mathbf{e}[n]\} = \mathbf{0}, \qquad \forall\ n \tag{4.2}$$

$$E\left\{\mathbf{e}[n_1]\mathbf{e}^{\mathtt{T}}[n_2]\right\} = \mathbf{0}, \qquad \forall\ n_1 \neq n_2. \tag{4.3}$$

$$E\left\{\mathbf{e}[n]\mathbf{e}^{\mathtt{T}}[n]\right\} = \mathbf{\Sigma}, \qquad \forall\ n \tag{4.4}$$

where $\mathbf{\Sigma}$ is a strictly positive definite matrix. Defining

$$\tilde{\mathbf{A}} \equiv \begin{bmatrix} \boldsymbol{\delta} & \mathbf{A}_1 & \mathbf{A}_2 & ... & \mathbf{A}_P \end{bmatrix},$$

$$\mathbf{y}[n] \equiv \begin{bmatrix} 1 & \mathbf{x}^{\mathtt{T}}[n-1] & \mathbf{x}^{\mathtt{T}}[n-2] & ... & \mathbf{x}^{\mathtt{T}}[n-P] \end{bmatrix}^{\mathtt{T}},$$

where $\mathbf{x}[n] = 0$ whenever $n < 0$, equation (4.1) reduces to

$$\mathbf{x}[n] = \tilde{\mathbf{A}}\mathbf{y}[n] + \mathbf{e}[n], \qquad n = 1, 2, ..., N. \tag{4.5}$$

Thus, a VAR model is parameterized by the regression coefficient matrix $\tilde{\mathbf{A}}$ and the innovation covariance matrix $\mathbf{\Sigma}$. The sum of squared error is given by

$$\begin{aligned} \mathcal{E} &= \sum_{n=1}^{N} \mathbf{e}^{\mathtt{T}}[n]\mathbf{e}[n] \\ &= \sum_{n=1}^{N} (\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n])^{\mathtt{T}} (\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n]), \end{aligned} \tag{4.6}$$

The ordinary least squares (OLS) estimate of $\tilde{\mathbf{A}}$ is obtained by differentiating the above equation with respect to $\tilde{\mathbf{A}}$ and equating the result to zero, yielding

$$\tilde{\mathbf{A}}_{OLS} = \left(\sum_{n=1}^{N} \mathbf{x}[n]\mathbf{y}^{\mathtt{T}}[n] \right) \left(\sum_{n=1}^{N} \mathbf{y}[n]\mathbf{y}^{\mathtt{T}}[n] \right)^{-1}, \tag{4.7}$$

The OLS estimate of the covariance matrix of the residual vectors, $\hat{\mathbf{\Sigma}}_{OLS}$, is given by [51]

$$\begin{aligned} \hat{\mathbf{\Sigma}}_{OLS} &= \frac{1}{N - DP - 1} \sum_{n=1}^{N} \hat{\mathbf{e}}[n]\hat{\mathbf{e}}^{\mathtt{T}}[n] \\ &= \frac{1}{N - DP - 1} \sum_{n=1}^{N} (\mathbf{x}[n] - \tilde{\mathbf{A}}_{OLS}\mathbf{y}[n])(\mathbf{x}[n] - \tilde{\mathbf{A}}_{OLS}\mathbf{y}[n])^{\mathtt{T}}. \end{aligned} \tag{4.8}$$

The OLS and the ML parameter estimates of $\tilde{\mathbf{A}}$ are equal when the distribution of the innovation sequence is Gaussian [51]. This is proven by noting that the likelihood function is given by

$$\mathrm{p}\big(\{\mathbf{x}[n]\}_{n=1}^{N}|\lambda\big) = \frac{1}{(2\pi)^{ND/2}|\boldsymbol{\Sigma}|^{N/2}} \exp\left(-\frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}[n]-\tilde{\mathbf{A}}\mathbf{y}[n])^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}[n]-\tilde{\mathbf{A}}\mathbf{y}[n])\right). \tag{4.9}$$

Differentiating the above equation with respect to $\tilde{\mathbf{A}}$ and equating the result to zero, expression (4.7) will follow. However, the ML estimate of $\boldsymbol{\Sigma}$ is a bit different from its OLS estimate viz

$$\hat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{N}\sum_{n=1}^{N}\hat{\mathbf{e}}[n]\hat{\mathbf{e}}^{\mathsf{T}}[n] = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}[n]-\hat{\tilde{\mathbf{A}}}_{OLS}\mathbf{y}[n])(\mathbf{x}[n]-\hat{\tilde{\mathbf{A}}}_{OLS}\mathbf{y}[n])^{\mathsf{T}}. \tag{4.10}$$

4.2 Parameter estimation of the VARGM model

As noted from the above subsection, the ML and the OLS parameter estimates of the autoregression matrix are equivalent only when the distribution of the innovation is Gaussian. However, this assumption may be restrictive in many applications. Alternatively, we may assume that the residual vectors follow the GMM distribution, i.e.,

$$\mathrm{p}(\mathbf{e}[n]) = \sum_{m=1}^{M} w_m \mathbb{N}(\mathbf{e}[n]; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m). \tag{4.11}$$

In this case, the intercept vector $\boldsymbol{\delta}$ can be dropped since the mean of the GMM distribution may well be different from zero.

In this case, the ML-estimate of the autoregression matrices, $\tilde{\mathbf{A}}$, will be different from the corresponding OLS-estimate. While the latter is still given by (4.7), the former is obtained through iterative procedures such as the gradient-based methods or the EM algorithm.

The problem of estimating the model parameters of a scalar autoregressive Gaussian mixture model via the EM algorithm was first investigated by Verbout et. al [120]. Basically, they proposed the EMAX algorithm as an iterative procedure for

estimating the model parameters. In this chapter, we generalize their procedure to deal with multivariate time series. In addition, more than one time series sequence may be used for parameter estimation. Having established such a generalization, the VARGM model will be more suitable for classification. In this section, we first consider the general case without any assumptions about the structures of the autoregression matrices or the covariance matrices. The special case of diagonal autoregression matrices and diagonal noise covariance matrices is addressed later.

4.2.1 The general case

Formally, there are K time series realizations, $\{\mathbf{x}_k[n]\}_{n=1}^{N_k}, k = 1, 2, ..., K$, the k^{th} of which contains N_k samples. All realizations are to be modeled by the following relation

$$\mathbf{x}_k[n] = \tilde{\mathbf{A}}\mathbf{y}_k[n-i] + \mathbf{e}_k[n], \qquad n = 1, 2, ..., N_k, \qquad k = 1, 2, ..., K \tag{4.12}$$

where $\mathbf{e}_k[n]$ follows the GMM distribution defined in (4.11) and

$$\tilde{\mathbf{A}} \equiv \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & ... & \mathbf{A}_P \end{bmatrix},$$

$$\mathbf{y}_k[n] \equiv \begin{bmatrix} \mathbf{x}_k^{\mathtt{T}}[n-1] & \mathbf{x}_k^{\mathtt{T}}[n-2] & ... & \mathbf{x}_k^{\mathtt{T}}[n-P] \end{bmatrix}^{\mathtt{T}}.$$

For convenience, the set of all time series sequences will be denoted by X. Assuming that each sequence is generated independently from other series, the likelihood of all sequences is equal to the product of the likelihood of the individual realizations, i.e.,

$$\begin{aligned} \mathrm{p}(X|\lambda) &= \prod_{k=1}^{K}\prod_{n=1}^{N_k} \mathrm{p}\Big(\mathbf{x}_k[n]\Big|\mathbf{x}_k[1:n-1], \lambda\Big) \\ &= \prod_{k=1}^{K}\prod_{n=1}^{N_k} \mathrm{p}\Big(\mathbf{x}_k[n]\Big|\mathbf{x}_k[n-P:n-1], \lambda\Big) \\ &= \prod_{k=1}^{K}\prod_{n=1}^{N_k} \left(\sum_{m=1}^{M} w_m \mathbb{N}(\mathbf{x}_k[n]; \tilde{\mathbf{A}}\mathbf{y}_k[n] + \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\right). \end{aligned} \tag{4.13}$$

Similar to the case of GMM, the new expression of the likelihood function in this case is so complex that direct differentiation leads to a set of highly nonlinear equations in the model parameters, which is extremely difficult to solve. Therefore, the EM algorithm is used again to estimate the model parameters. The complete data specification is $Z = \{X, \Phi\}$, where $\Phi = \{\{\phi_k[n]\}_{n=1}^{N_k}\}_{k=1}^{K}$ and $\phi_k[n]$ is the index of the Gaussian component selected at instant n for time series realization k. The complete likelihood function is given by

$$\begin{aligned}
\mathrm{p}(X, \Phi|\lambda) &= \prod_{k=1}^{K}\prod_{n=1}^{N_k} \mathrm{p}(\mathbf{x}_k[n], \phi_k[n]|\mathbf{x}_k[1:n-1], \phi_k[1:n-1], \lambda) \\
&= \prod_{k=1}^{K}\prod_{n=1}^{N_k} P(\phi_k[n]|\lambda)\mathrm{p}(\mathbf{x}_k[n]|\mathbf{x}_k[n-P:n-1], \phi_k[n], \lambda) \\
&= \prod_{k=1}^{K}\prod_{n=1}^{N_k} w_{\phi_k[n]}\mathbb{N}(\mathbf{x}_k[n]; \tilde{\mathbf{A}}\mathbf{y}_k[n] + \boldsymbol{\mu}_{\phi[n]}, \boldsymbol{\Sigma}_{\phi[n]}). \qquad (4.14)
\end{aligned}$$

Similar to the GMM case, the VARGM model parameters are iteratively updated by maximizing the following auxiliary function.

$$Q(\lambda; X, \lambda^{(s)}) = E\left\{\log P\left(X, \Phi|\lambda\right)|X, \lambda^{(s)}\right\}, \qquad (4.15)$$

where $\lambda^{(s)}$ is the set of model parameters obtained after the s^{th} iteration. Substituting (4.14) into (4.15), and simplifying, the auxiliary function is given by

$$\begin{aligned}
Q(\lambda; X, \lambda^{(s)}) = c + \sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \Bigg(&\log w_m - \frac{1}{2}\log|\boldsymbol{\Sigma}_m| \\
&-\frac{1}{2}(\mathbf{x}_k[n] - \tilde{\mathbf{A}}\mathbf{y}_k[n] - \boldsymbol{\mu}_m)^{\mathrm{T}}\boldsymbol{\Sigma}_m^{-1}(\mathbf{x}_k[n] - \tilde{\mathbf{A}}\mathbf{y}_k[n] - \boldsymbol{\mu}_m)\Bigg), \qquad (4.16)
\end{aligned}$$

where c does not depend on the model parameters, $\sum_{k,n,m}$ is a shorthand for $\sum_{k=1}^{K}\sum_{n=1}^{N_k}\sum_{m=1}^{M}$, $P_{k,n,m}(\lambda^{(s)})$ is the a posteriori probability of the m^{th} Gaussian component given the observation $\mathbf{x}_k[n]$, i.e.,

$$\begin{aligned}
P_{k,n,m}(\lambda^{(s)}) &\equiv P\left(\phi_k[n] = m \,\middle|X; \lambda^{(s)}\right) \\
&= \frac{w_m^{(s)}\mathbb{N}(\mathbf{x}_k[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}_k[n]; \boldsymbol{\mu}_m^{(s)}, \boldsymbol{\Sigma}_m^{(s)})}{\sum_{m'=1}^{M} w_{m'}^{(s)}\mathbb{N}(\mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}_k[n]; \boldsymbol{\mu}_{m'}^{(s)}, \boldsymbol{\Sigma}_{m'}^{(s)})}. \qquad (4.17)
\end{aligned}$$

The following update equation for the prior probabilities can be obtained in a similar way to that followed in Chapter 3.

$$w_m^{(s+1)} = \frac{1}{\sum_{k=1}^{K} N_k} \sum_{k=1}^{K} \sum_{n=1}^{N_k} P_{k,n,m}(\lambda^{(s)}) \tag{4.18}$$

The update equations of $\boldsymbol{\mu}_m^{(s+1)}$, $\boldsymbol{\Sigma}_m^{(s+1)}$, and $\tilde{\mathbf{A}}^{(s+1)}$ are obtained by equating the derivative of (4.16) with respect to $\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$, and $\tilde{\mathbf{A}}$ to zero. Similar to the scalar case in [120], this results in a set of nonlinear equations that are difficult to solve simultaneously. Instead, at each iteration, only one variable is maximized while others are kept fixed. This approach guarantees coordinate ascent convergence to a local maximum [74]. Therefore, we start by finding optimum $\boldsymbol{\mu}_m$ while keeping other variables constant, then finding optimum $\boldsymbol{\Sigma}_m$ then optimum $\tilde{\mathbf{A}}$. After some simple manipulation, the following update equations are obtained.

$$\boldsymbol{\mu}_m^{(s+1)} = \frac{\sum_{k=1}^{K} \sum_{n=1}^{N_k} P_{k,n,m}(\lambda^{(s)})(\mathbf{x}_k[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}_k[n])}{\sum_{k=1}^{K} \sum_{n=1}^{N_k} P_{k,n,m}(\lambda^{(s)})} \tag{4.19}$$

$$\begin{aligned}\boldsymbol{\Sigma}_m^{(s+1)} = & \frac{\sum_{k=1}^{K} \sum_{n=1}^{N_k} P_{k,n,m}(\lambda^{(s)})(\mathbf{x}_k[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}_k[n])(\mathbf{x}_k[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}_k[n])^{\mathsf{T}}}{\sum_{k=1}^{K} \sum_{n=1}^{N_k} P_{k,n,m}(\lambda^{(s)})} \\ & - \boldsymbol{\mu}_m^{(s+1)}(\boldsymbol{\mu}_m^{(s+1)})^{\mathsf{T}}\end{aligned} \tag{4.20}$$

The update equation of $\tilde{\mathbf{A}}$ deserves some investigation. Equating the derivative of $Q(\lambda; X, \lambda^{(s)}))$ to zero, the following equation in $\tilde{\mathbf{A}}$ is obtained

$$\begin{aligned}&\sum_{k,n,m} P_{k,n,m}(\lambda^{(s)})(\boldsymbol{\Sigma}_m^{-1})^{(s+1)}\tilde{\mathbf{A}}^{(s+1)}\mathbf{y}_k[n]\mathbf{y}_k^{\mathsf{T}}[n] \\ &= \sum_{k,n,m} P_{k,n,m}(\lambda^{(s)})(\boldsymbol{\Sigma}_m^{-1})^{(s+1)}(\mathbf{x}_k[n] - \boldsymbol{\mu}_m^{(s+1)})\mathbf{y}_k^{\mathsf{T}}[n],\end{aligned} \tag{4.21}$$

Applying the $\mathsf{vec}\,()$ operator to both sides of the above equation, the following expression for $\mathsf{vec}\left(\tilde{\mathbf{A}}^{(s+1)}\right)$ could be obtained after some simple manipulations.

$$\begin{aligned}\mathsf{vec}\left(\tilde{\mathbf{A}}^{(s+1)}\right) = & \left[\sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \left((\mathbf{y}_k[n]\mathbf{y}_k^{\mathsf{T}}[n]) \otimes (\boldsymbol{\Sigma}_m^{-1})^{(s+1)}\right)\right]^{-1} \times \\ & \mathsf{vec}\left(\sum_{k,n,m} P_{k,n,m}(\lambda^{(s)})(\boldsymbol{\Sigma}_m^{-1})^{(s+1)}(\mathbf{x}_k[n] - \boldsymbol{\mu}_m^{(s+1)})\mathbf{y}_k^{\mathsf{T}}[n]\right),\end{aligned} \tag{4.22}$$

where $\otimes$ denotes the *Kronecker product* of two matrices.

However, the space required to store the matrix to be inverted in (4.22) is of order $\mathcal{O}(D^4P^2)$. For some applications, this may require excessive storage. An alternative way for obtaining $\tilde{\mathbf{A}}^{(s+1)}$ is through iterative methods. Fortunately, the auxiliary function in (4.16) is quadratic in $\tilde{\mathbf{A}}$. Hence, it has a unique maximizer, which is also the unique solution of (4.21). Thus, fast and efficient techniques such as the steepest ascent method and the conjugate gradient method may be used to estimate this unique maximizer. In this thesis, general VARGM were applied only to the speech emotion classification problem and the memory requirement of the EM algorithm was reasonable. Therefore, we did not apply any approximation to (4.22). For the speaker identification problem, we prefer to use VARGM models with diagonal autoregression matrices and diagonal covariance matrices.

4.2.2 Diagonal autoregression matrices

Another method to reduce the computational complexity of the training and the testing algorithms is to assume that the autoregression matrices, $\mathbf{A}_1, \ldots, \mathbf{A}_P$ as well as the covariance matrices, $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_M$, are diagonal. Though it is also mathematically tractable to consider other cases, e.g. general covariance matrices and diagonal autoregression matrices, no significant advantage was experimentally observed with this assumption. Therefore, only the case of diagonal autoregression matrices and diagonal covariance matrices will be covered in this thesis.

In order to simplify our derivations, it is convenient to define the following two vectors:

- $\tilde{\mathbf{a}}_d = \begin{bmatrix} \tilde{a}_{d1} & \ldots & \tilde{a}_{dP} \end{bmatrix}^{\mathsf{T}}$, where $\tilde{a}_{dp}$ is the d^{th} on the diagonal of $\mathbf{A}_p$.
- $\tilde{\mathbf{y}}_{k,d}[n] = \begin{bmatrix} y_d[n-1] & \ldots & y_d[n-P] \end{bmatrix}^{\mathsf{T}}$

For the case of diagonal covariance matrices and diagonal autoregression matrices,

it is straightforward to show that the auxiliary function in (4.16) simplifies to

$$Q(\lambda; X, \lambda^{(s)}) = \\ c + \sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \left(\log w_m - \frac{1}{2} \sum_{d=1}^{D} \left(\log \sigma_{m,d}^2 + \frac{(x_{k,d}[n] - \tilde{\mathbf{a}}_d^{\mathsf{T}} \tilde{\mathbf{y}}_{k,d}[n] - \mu_{m,d})^2}{\sigma_{m,d}^2} \right) \right), \tag{4.23}$$

where d in the subscript refers to the d^{th} components of the vector. Similar to the general case, we have the same constraint on the prior probabilities. Thus, the parameter update equations are obtained by following the same steps used in the general case. The only difference is that (4.16) is replaced by (4.23). It is not hard to show that the update equations for the prior probabilities and the mean vectors are still given by (4.18) and (4.19), respectively. The update equations for $\sigma_{m,d}^2$ and $\tilde{\mathbf{a}}_d$ are given by

$$(\sigma_{m,d}^{(s+1)})^2 = \frac{\sum_{k,n} P_{k,n,m}(\lambda^{(s)})(x_{k,d}[n] - (\tilde{\mathbf{a}}_d^{(s)})^{\mathsf{T}} \tilde{\mathbf{y}}_{k,d}[n] - \mu_{m,d}^{(s)})^2}{\sum_{k,n} P_{k,n,m}(\lambda^{(s)})} \tag{4.24}$$

$$\tilde{\mathbf{a}}_d^{(s+1)} = \left(\sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \frac{\tilde{\mathbf{y}}_{k,d}[n] \tilde{\mathbf{y}}_{k,d}^{\mathsf{T}}[n]}{(\sigma_{m,d}^{(s+1)})^2} \right)^{-1} \left(\sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \frac{\mathbf{x}_{k,d}[n] - (\mu_{m,d}^{(s)})^2}{(\sigma_{m,d}^{(s+1)})^2} \right) \tag{4.25}$$

The above update equations require a space of order $\mathcal{O}(P^2)$ for each vector $\tilde{\mathbf{a}}_d$, which is significantly less than that required for the general case.

4.3 Model order selection

Similar to other classifiers, the VARGM model contains two types of parameters: numeric parameters that are estimated using the ML or the MAP estimation criteria and structural design parameters that control the classifier complexity. In our proposed VARGM classifier, the structural design parameters are the regression order, P, and the number of Gaussian components, M, in the distribution of the residual vectors. In this context, a model order refers to a specific combination (M, P). Unfortunately, there is no straightforward way to determine the optimum

model order that provide a VARGM classifier with the optimal generalization ability. In general, structural design parameters are estimated either by trial and error methods or according to a certain model order selection criterion. In the latter approach, the model order selection criterion is calculated for different orders. The selected order is the one corresponding to the minimum value. There are two basic types of model order selection criteria: the cross validation error [13] and the information-theoretic criteria.

In the cross validation method, the training data is divided into two sets: one for estimating the classifier parameters and the other for validating its performance. The model selection criterion is taken as the classification error with respect to the validation set. The k-fold cross validation can be incorporated easily into this method in order to get more reliable estimates of the validation error. This method is useful when the amount of the training and the testing data is not sufficient for a reliable estimate of the model parameters.

Information-theoretic criteria, on the other hand, attempt to find the best possible compromise between the classifier ability to adequately model the distribution of the training data and the complexity of the classifier. Usually, information theoretic criteria are given in the following form

$$IC(M,P) = -\alpha \log \mathrm{p}(X|\lambda(M,P)) + \beta|\lambda(M,P)|, \tag{4.26}$$

where α and β are some positive constants and $\lambda(M,P)$ denotes a VARGM model with M Gaussian components and regression order P. $|\lambda(M,P)|$ is the number of parameters in $\lambda(M,P)$. The first term in (4.26) measures the fitness of the data to the distribution specified by the classifier model and the second term measures the complexity of the classifier. In this thesis, the following information-theoretic criteria are considered

1. Akaike information criterion (AIC) [1]

$$AIC(M,P) = -2 \log \mathrm{p}(X|\lambda(M,P)) + 2|\lambda(M,P)| \tag{4.27}$$

2. Kullback information criterion (KIC)[19]

$$KIC(P) = -2\log \mathrm{p}(X|\lambda(M,P)) + 3|\lambda(M,P)| \tag{4.28}$$

3. Bayesian information criterion (BIC)[101]

$$BIC(P) = -2\log \mathrm{p}(X|\lambda(M,P)) + |\lambda(M,P)|\log(N) \tag{4.29}$$

For our VARGM model, the likelihood function is given by (4.13) and the number of model parameters is given by

$$|\lambda(M,P)| = PD^2 + M(1 + D + D(D+1)/2) \tag{4.30}$$

for VARGM models with full covariance matrices and full auto-regression matrices, and

$$|\lambda(M,P)| = PD + M(1 + 2D) \tag{4.31}$$

for VARGM models with diagonal covariance matrices and diagonal auto-regression matrices.

The standard information theoretic model order selection technique may be time consuming since the first term in (4.26) requires training models with different orders. Moreover, for each model order, initialization should be done properly in order to ensure that the likelihood function increases with the increase of the M or P.

First, let us consider the search of the optimal regression order assuming that M is known. The set of all VARGM models with regression order $P-1$ is a subset of all VARGM models with regression order P. This statement is established easily by setting $\mathbf{A}_P = \mathbf{0}_{D\times D}$ in any model with regression order P. Thus, for a given M, Algorithm 4.1 can be used to calculate the likelihood function for different regression orders.

It should be mentioned that only few iterations are required for the EM algorithm in the above algorithm since the increase in the likelihood function is

Algorithm 4.1 Selection of the regression order.

1: **Inputs:** X, M, P_{max}, and $\lambda(M, 0)$.
2: **Output:** $\lambda(M, P)$, $P = 1, \ldots, P_{max}$.
3: Fit the data into a GMM with M Gaussian components.
4: Calculate $\mathrm{p}(X|\lambda(M, 0))$.
5: Calculate $IC(M, 0)$ using (4.26).
6: **for** $P = 1$ to P_{max} **do**
7: Set $\lambda_0 = \lambda(M, P-1)$ (Copy all the parameters of $\lambda(M, P-1)$ to λ_0).
8: Increment the regression order of λ_0 by 1.
9: In λ_0, put $\mathbf{A}_P = \mathbf{0}_{D\times D}$.
10: Train $\lambda(M, P)$ using the EM algorithm. Take λ_0 as the initial model for the EM algorithm.
11: Calculate $\mathrm{p}(X|\lambda(M, P))$ using (4.13).
12: Calculate $IC(M, P)$ using (4.26).
13: **end for**

guaranteed. In addition, it is unlikely that having a small number of iterations will affect the choice of the optimum model order.

The question now is how to determine the optimum number of Gaussian components. For speaker identification and verification problems, it is a common practice to select M as a power of 2 [103]. In this work, we prefer to follow this practice for two reasons. First, it is useful to apply mixture splitting to speed up model order selection, as we shall see shortly. Second, significant difference in the classification accuracy is only observed when the change in the number of Gaussian components is relatively large. A possible binary split procedure is shown in Algorithm 4.2.

Thus, our two dimensional search for the optimal combination (M, P) proceeds as follows. First, we consider GMMs ($P = 0$). We calculate the model selection criterion for different values of M. The binary split algorithm, shown in Algorithm 4.2, is used to speed up the estimation process of the incomplete likelihood function. For each value of M, we calculate the model selection criterion for different

Algorithm 4.2 The binary split algorithm.

1: **Inputs:** X, M, and $\lambda(M, 0)$.

2: **Output:** $\lambda(2M, 0)$.

3: Generate M uniformly distributed random numbers in the range from 0 to 1.

4: Normalize the generated numbers so that they sum to one. These numbers are the initial priors of new M components, $\hat{w}_1, \ldots, \hat{w}_M$.

5: Generate random vectors $\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_M$. These are the initial centers of the new M components.

6: Initialize $\lambda(2M, 0)$ with the following parameters.

1.
$$\tilde{w}_m = \begin{cases} (1-\epsilon)w_m & m = 1, \ldots, M. \\ \epsilon\hat{w}_m & m = M+1, \ldots, 2M, \end{cases}$$
where ϵ is a small quantity.

2.
$$\tilde{\boldsymbol{\mu}}_m = \begin{cases} \boldsymbol{\mu}_m & m = 1, \ldots, M. \\ \tilde{\boldsymbol{\mu}}_m & m = M+1, \ldots, 2M. \end{cases}$$

3.
$$\tilde{\boldsymbol{\Gamma}}_m = \begin{cases} \boldsymbol{\Gamma}_m & m = 1, \ldots, M. \\ \mathbf{I}_D & m = M+1, \ldots, 2M. \end{cases}$$

7: Update the model parameters using the k-means algorithm then the EM algorithm.

regression orders as shown in Algorithm 4.1. The overall model order selection procedure is shown in Algorithm 4.3.

Algorithm 4.3 The proposed model order selection algorithm

1: **Inputs:** X.
2: **Output:** Optimal M and P.
3: Fit the data to $\lambda(M_0, 0)$, where $M_0 = 2^{m_0}$ is an initial estimate for the model order and m_0 is an integer.
4: Calculate $\mathrm{p}(X|\lambda(M_0, 0))$ using (4.13).
5: Calculate $IC(M_0, 0)$ using (4.26).
6: **for** $m = m_0 + 1$ to m_{max} **do**
7: $M = 2^m$.
8: Use Algorithm 4.2 to split the components in $\lambda(M/2, 0)$. The resultant model is $\lambda(M, 0)$.
9: Calculate $\mathrm{p}(X|\lambda(M, 0))$ using (4.13).
10: Calculate $IC(M, 0)$ using (4.26).
11: **end for**
12: **for** $m = m_0$ to m_{max} **do**
13: $M = 2^m$.
14: **for** $P = 1$ to P_{max} **do**
15: Use Algorithm 4.1 to derive $\lambda(M, P)$ from $\lambda(M, P - 1)$.
16: Calculate $\mathrm{p}(X|\lambda(M, P))$ using (4.13).
17: Calculate $IC(M, P)$ using (4.26).
18: **end for**
19: **end for**
20: The selected order (M, P) is given by

$$\{M^*, P^*\} = \arg\min_{M,P} IC(M, P).$$

4.4 Classification using the VARGM model

In principle, the classification methodology using the VARGM model is very similar to that of the GMM. Therefore, in this section, we just point out the main differences between the GMM-based and the corresponding VARGM-based frameworks.

4.4.1 Standard VARGM/ML framework

The training procedure in the standard VARGM/ML-based framework is similar to that illustrated in figure 3.1. In the training phase, the only difference is that both the model order selection procedure and the parameter estimation procedure can be integrated together as described in Algorithm 4.3 in section 4.3. Hence, this algorithm should replace the parameter estimation module in figure 3.1. In the classification phase, speakers' scores are calculated using equation (4.13) instead of (3.1).

4.4.2 VARGM/UBM

In the VARGM/UBM framework, all the training data vectors are used to estimate the UBM parameters as done with the GMM case. For adapting individual speaker model parameters, we follow the same methodology employed in section 3.3. For simplicity, we shall consider only the case of a single sequence for adaptation, i.e. $K = 1$. Nonetheless, the general case is rather straightforward. For general autoregression matrices, the prior density of $\tilde{\mathbf{A}}$ is assumed in the following form.

$$\mathrm{p}\Big(\tilde{\mathbf{A}}|\beta, \tilde{\mathbf{A}}_{\mathrm{UBM}}\Big) \propto \exp\left(-\frac{\beta}{2}\mathrm{trace}\left((\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_{\mathrm{UBM}})^{\mathsf{T}}(\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_{\mathrm{UBM}})\right)\right), \tag{4.32}$$

where β is an update factor for the autoregression matrix and $\tilde{\mathbf{A}}_{\mathrm{UBM}}$ is the autoregression matrix of the UBM. For diagonal autoregression matrices, the prior density of the autoregression matrices are assumed to be in the form

$$\mathrm{p}(\tilde{\mathbf{a}}_d|\tilde{\mathbf{a}}_{d,\mathrm{UBM}}) \propto \exp\left(-\frac{\beta}{2}(\tilde{\mathbf{a}}_d - \tilde{\mathbf{a}}_{d,\mathrm{UBM}})^{\mathsf{T}}(\tilde{\mathbf{a}}_d - \tilde{\mathbf{a}}_{d,\mathrm{UBM}})\right), \tag{4.33}$$

where $\tilde{\mathbf{a}}_d$ is as defined in subsection 4.2.2.

The update equations are derived in a very similar way to that followed in section 3.3. Hence, we state them without derivations. For general covariance and autoregression matrices, we have the following update equations.

$$w_m^{(s+1)} = \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \nu_m - 1}{N - M + \sum_{m=1}^{M} \nu_m} \tag{4.34}$$

$$\boldsymbol{\mu}_m^{(s+1)} = \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) \mathbf{e}^{(s)}[n] + \tau_m \tilde{\boldsymbol{\mu}}_m}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \tau_m} \tag{4.35}$$

$$\begin{aligned} \boldsymbol{\Sigma}_m^{(s+1)} = & \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) \mathbf{e}^{(s)}[n](\mathbf{e}^{(s)}[n])^{\mathsf{T}} + \tau_m(\tilde{\boldsymbol{\mu}}_m \tilde{\boldsymbol{\mu}}_m^{\mathsf{T}} + \tilde{\boldsymbol{\Sigma}}_m)}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \alpha_m - d - 1} \\ & - \frac{\left(\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \tau_m\right) \boldsymbol{\mu}_m^{(s+1)} (\boldsymbol{\mu}_m^{(s+1)})^{\mathsf{T}}}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \alpha_m - d - 1} \end{aligned} \tag{4.36}$$

$$\begin{aligned} \mathsf{vec}\left(\tilde{\mathbf{A}}^{(s+1)}\right) = & \left[\beta \mathbf{I} + \sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \left((\mathbf{y}_k[n]\mathbf{y}_k^{\mathsf{T}}[n])(\boldsymbol{\Sigma}_m^{-1})^{(s+1)}\right)\right]^{-1} \times \\ & \mathsf{vec}\left(\beta \tilde{\mathbf{A}}_{\text{UBM}} + \sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) (\boldsymbol{\Sigma}_m^{-1})^{(s+1)} (\mathbf{x}_k[n] - \boldsymbol{\mu}_m^{(s+1)}) \mathbf{y}_k^{\mathsf{T}}[n]\right), \end{aligned} \tag{4.37}$$

where

$$\mathbf{e}^{(s)}[n] = \mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)} \mathbf{y}[n].$$

For the case of diagonal covariance matrices and diagonal autoregression matrices, equations (4.36) and (4.37) simplify to

$$\begin{aligned} (\sigma_d^2)^{(s+1)} = & \frac{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) (e_d^{(s)})^2[n] + \tau_m(\tilde{\mu}_{m,d}^2 + \tilde{\sigma}_{m,d}^2)}{\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \alpha_m - d - 1} \\ & - \frac{\left(\sum_{n=1}^{N} P_{m,n}(\lambda^{(s)}) + \tau_m\right) (\mu_{m,d}^{(s+1)})^2}{\lambda^{(s)}) + \alpha_m - d - 1}. \end{aligned} \tag{4.38}$$

and

$$\begin{aligned} \tilde{\mathbf{a}}_d^{(s+1)} = & \left(\beta \mathbf{I} + \sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \frac{\tilde{\mathbf{y}}_{k,d}[n] \tilde{\mathbf{y}}_{k,d}^{\mathsf{T}}[n]}{(\sigma_{m,d}^{(s+1)})^2}\right)^{-1} \times \\ & \left(\beta \mathbf{a}_{d,\text{UBM}} + \sum_{k,n,m} P_{k,n,m}(\lambda^{(s)}) \frac{\mathbf{x}_{k,d}[n] - (\mu_{m,d}^{(s)})^2}{(\sigma_{m,d}^{(s+1)})^2}\right) \end{aligned} \tag{4.39}$$

Rather than the differences in the update equations, the training and the classification methodologies are the same.

Chapter 5

Generalized maximum likelihood adaptation

In this chapter, we basically describe our proposed GML adaptation technique. We consider a particular scenario where the training environment is distortion free but the testing environment is corrupted by band limitation and additive noise with a partially unknown distribution. Similar to the PCM method, we assume a general model for the testing feature vectors, which contains both clean speech parameters and noise parameters. While the clean speech parameters are estimated in the training phase, the distortion parameters are estimated from the feature vectors extracted from corrupted speech in the testing phase. After estimation of the distortion parameters, the ML decision rule is applied in order to determine the most likely speaker of the testing utterance. In Chapter 6, we shall show that such a compensation technique results in an increased robustness of the speaker identification systems against additive and convolutive noise.

This chapter is divided into four sections. The main statistical model for distorted speech is described in section 5.1. In section 5.2, we explain how to estimate the distortion related parameters from the given speech. In section 5.3, we modify the model order selection technique, proposed in section 4.3, so that it fits into our GML adaptation framework. Moreover, we propose another approximate but

faster version of this model order selection algorithm. A global picture of the GML adaptation technique is illustrated in section 5.4. Finally, in section 5.5, we present a potential application to this GML compensation technique, namely blind equalization of MIMO channels [7].

5.1 Main statistical model

Generally a variety of distortion models may be assumed. Typically, the most two important factors that hamper the classification performance are the spectral distortion caused by the communication channel and the additive noise contaminating the speech signal. The former factor is equivalent to bandpass filtering effect. Modeling the band-limitation as infinite impulse response (IIR) filter, we have the following distortion model for the extracted testing feature vectors, $\mathbf{x}[1:N]$.

$$\mathbf{x}[n] = \sum_{p=1}^{P} \mathbf{A}_{p,s}\mathbf{x}[n-p] + \mathbf{s}[n] + \mathbf{e}[n], \tag{5.1}$$

where $\mathbf{s}[n]$ refers to a hypothetical feature vector extracted from the clean part of the speech signal and $\mathbf{e}[n]$ corresponds to the noisy part of the speech. The second subscript s in $\mathbf{A}_{p,s}$ refers to the speaker index. The distribution of $\mathbf{s}[n]$ is assumed to be a GMM whose parameters are estimated clean training speech; i.e.,

$$\mathrm{p}(\mathbf{s}[n]|\mathcal{H}_s) = \sum_{m=1}^{M} w_{m,s}\mathbb{N}(\mathbf{s}[n]; \boldsymbol{\mu}_{m,s}, \boldsymbol{\Sigma}_{m,s}). \tag{5.2}$$

The noise vector $\mathbf{e}[n]$ follows a speaker-independent Gaussian distribution, $\mathbb{N}(\mathbf{e}[n]; \mathbf{0}, \boldsymbol{\Gamma})$, where $\boldsymbol{\Gamma}$ is assumed to be unknown. Hence, in the testing phase, the speaker-dependent parameters $\lambda_s = \{w_{m,s}, \mu_{m,s}, \boldsymbol{\Sigma}_{m,s}\}_{m=1}^{M}$, $s = 1, \ldots, S$ are considered known while the auto-regression matrices and the noise covariance matrices in (5.1) are estimated from the testing feature vectors. In the following sections, we shall denote the distortion parameters by θ.

5.2 Model parameter estimation

In this section, we first consider the parameter estimation problem for general autoregressive and covariance matrices given that one of the hypotheses $\mathcal{H}_1, \ldots, \mathcal{H}_S$ is true. In order to simplify the notation, we shall drop the speaker index s. It should be stated that the following parameter estimation procedure is repeated for each speaker model as will be illustrated in section 5.2. Other special structures of the autoregressive and the noise covariance matrices are addressed shortly. The likelihood function of the observed data sequence, $\mathbf{x}[1:N]$, is given by

$$\begin{aligned} \mathrm{p}\Big(\mathbf{x}[1:N]\Big|\theta\Big) &= \prod_{n=1}^{N} \mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{x}[1:n-1],\theta\Big) = \prod_{n=1}^{N} \mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{x}[n-P:n-1],\theta\Big) \\ &= \prod_{n=1}^{N}\sum_{m=1}^{M} w_m \mathbb{N}(\mathbf{x}[n]; \tilde{\mathbf{A}}\mathbf{y}[n] + \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m + \boldsymbol{\Gamma}), \end{aligned} \tag{5.3}$$

where $\tilde{\mathbf{A}} = \Big[\mathbf{A}_1 \quad \ldots \quad \mathbf{A}_P\Big]$ and $\mathbf{y}[n] = \Big[\mathbf{x}[n-1]^{\mathsf{T}} \quad \ldots \quad \mathbf{x}[n-P]^{\mathsf{T}}\Big]^{\mathsf{T}}$. It is clear that the likelihood function is highly nonlinear in $\tilde{\mathbf{A}}$ and $\boldsymbol{\Sigma}$. Similar to the estimation approaches in chapters 3 and 4, the iterative EM procedure is used to maximize the likelihood function with respect to $\tilde{\mathbf{A}}$ and $\boldsymbol{\Sigma}$.

Initial guess for the model parameters can be found by fitting the data to the following autoregressive model

$$\mathbf{x}[n] = \tilde{\mathbf{A}}\mathbf{y}[n] + \mathbf{e}[n], \tag{5.4}$$

where $\mathrm{p}(\mathbf{e}[n]) = \mathbb{N}(\mathbf{e}[n]; \mathbf{z}, \boldsymbol{\Sigma})$. The ML estimates for $\tilde{\mathbf{A}}$ and $\boldsymbol{\Sigma}$ given the above model are the initial values for the EM algorithm and are denoted by $\tilde{\mathbf{A}}^{(0)}$ and $\boldsymbol{\Sigma}^{(0)}$, respectively. It is straightforward to show that

$$\tilde{\mathbf{A}}^{(0)} = \left(\sum_{n=1}^{N} \mathbf{x}[n]\mathbf{y}[n]^{\mathsf{T}}\right)\left(\sum_{n=1}^{N} \mathbf{y}[n]\mathbf{y}[n]^{\mathsf{T}}\right)^{-1} \tag{5.5}$$

$$\boldsymbol{\Sigma}^{(0)} = \frac{1}{T}\sum_{n=1}^{N} (\mathbf{x}[n] - \tilde{\mathbf{A}}^{(0)}\mathbf{y}[n])(\mathbf{x}[n] - \tilde{\mathbf{A}}^{(0)}\mathbf{y}[n])^{\mathsf{T}} \tag{5.6}$$

For deriving the update equations, the complete data specification in this problem will include the clean data vectors $\mathbf{s}[n]$ as well as the index function $\phi[n]$,

defined before in subsection 3.2.1 will significantly simplify our derivations. Define $X = \{\mathbf{x}[1:N]\}$, $\Phi = \{\phi[1:N]\}$, and $S = \{\mathbf{s}[1:N]\}$. The complete log-likelihood function is given by

$$\begin{aligned}
&\log \mathrm{p}\Big(X, \Phi, S\Big|\theta\Big) \\
&= \sum_{n=1}^{N} \log \mathrm{p}\Big(\mathbf{x}[n], \phi[n], \mathbf{s}[n]\Big|\mathbf{x}[1:n-1], \phi[1:n-1], \mathbf{s}[1:n-1], \theta\Big) \\
&= \sum_{n=1}^{N} \Big(\log \mathrm{p}\Big(\phi[n]\Big|\theta\Big) + \log \mathrm{p}\Big(\mathbf{s}[n]\Big|\phi[n], \theta\Big) + \log \mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{s}[n], \mathbf{x}[n-P:n-1], \phi[n], \theta\Big)\Big) \\
&= \sum_{n=1}^{N} \Big(\log w_{\phi[n]} + \log \mathbb{N}(\mathbf{s}[n]; \boldsymbol{\mu}_{\phi[n]}, \boldsymbol{\Sigma}_{\phi[n]}) + \log \mathbb{N}(\mathbf{x}[n]; \tilde{\mathbf{A}}\mathbf{y}[n] + \mathbf{s}[n], \boldsymbol{\Gamma})\Big) \qquad (5.7)
\end{aligned}$$

Obviously, the first two terms in the above expression do not contain the parameters to be estimated: $\tilde{\mathbf{A}}$ and $\boldsymbol{\Gamma}$. Therefore, they can be ignored. Substituting (5.7) into (3.5), the following expression for the auxiliary function is obtained

$$\begin{aligned}
&Q(\theta; X, \theta^{(s)}) \\
&= c - \frac{1}{2}\sum_{n=1}^{N} \Big(\log|\boldsymbol{\Gamma}| + E\Big\{(\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n] - \mathbf{s}[n])^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}(\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n] - \mathbf{s}[n])\Big|\mathbf{x}[1:N], \theta^{(s)}\Big\}\Big), \qquad (5.8)
\end{aligned}$$

where c does not depend on the estimated parameters. Hence, in order to evaluate and optimize the auxiliary function $Q(\theta; X, \theta^{(s)})$, it is necessary to derive smoothed estimates of the first and the second order statistics of $\mathbf{s}[n]$ given all the noisy data $\mathbf{x}[1:N]$. In order to simplify the notations, define the following smoothed statistics

$$\hat{\mathbf{s}}[n] \equiv E\Big\{\mathbf{s}[n]\Big|\mathbf{x}[1:N], \theta^{(s)}\Big\}. \qquad (5.9)$$

$$\mathbf{R}[n] \equiv E\Big\{\mathbf{s}[n]\mathbf{s}^{\mathrm{T}}[n]\Big|\mathbf{x}[1:N], \theta^{(s)}\Big\}. \qquad (5.10)$$

$$\hat{\mathbf{s}}[n|m] \equiv E\Big\{\mathbf{s}[n]\Big|\phi[n] = m, \mathbf{x}[1:N], \theta^{(s)}\Big\} \qquad (5.11)$$

$$\mathbf{R}[n|m] \equiv E\Big\{\mathbf{s}[n]\mathbf{s}^{\mathrm{T}}[n]\Big|\phi[n] = m, \mathbf{x}[1:N], \theta^{(s)}\Big\}. \qquad (5.12)$$

In appendix A, we prove the following expressions for $\hat{\mathbf{s}}[n]$ and $\mathbf{R}[n]$.

$$\hat{\mathbf{s}}[n] = \sum_{m=1}^{M} P_{m,n}(\theta^{(s)})\hat{\mathbf{s}}[n|m] \tag{5.13}$$

$$\mathbf{R}[n] = \sum_{m=1}^{M} P_{m,n}(\theta^{(s)}) \left(\mathbf{R}[n|m] + \hat{\mathbf{s}}[n|m]\hat{\mathbf{s}}^{\mathsf{T}}[n|m]\right) - \hat{\mathbf{s}}[n]\hat{\mathbf{s}}^{\mathsf{T}}[n], \tag{5.14}$$

where

$$\hat{\mathbf{s}}[n|m] = \boldsymbol{\mu}_m^{(s)} + \mathbf{K}_m(\mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}[n] - \boldsymbol{\mu}_m^{(s)}) \tag{5.15}$$

$$\mathbf{R}[n|m] = (\mathbf{I} - \mathbf{K}_m)\boldsymbol{\Sigma}_m^{(s)} \tag{5.16}$$

$$\mathbf{K}_m = \boldsymbol{\Sigma}_m^{(s)}(\boldsymbol{\Gamma} + \boldsymbol{\Sigma}_m^{(s)})^{-1}.$$

$$P_{m,n}(\theta^{(s)}) = \frac{w_m^{(s)}\mathbb{N}(\mathbf{x}[n]; \tilde{\mathbf{A}}^{(s)}\mathbf{y}[n] + \boldsymbol{\mu}_m^{(s)}, \boldsymbol{\Sigma}_m^{(s)} + \boldsymbol{\Gamma}^{(s)})}{\sum_{m'=1}^{M} w_{m'}^{(s)}\mathbb{N}(\mathbf{x}[n]; \tilde{\mathbf{A}}^{(s)}\mathbf{y}[n] + \boldsymbol{\mu}_{m'}^{(s)}, \boldsymbol{\Sigma}_{m'}^{(s)} + \boldsymbol{\Gamma}^{(s)})}$$

The update equations for $\tilde{\mathbf{A}}$ and $\boldsymbol{\Gamma}$ are obtained by differentiating the auxiliary function with respect to $\tilde{\mathbf{A}}$ and $\boldsymbol{\Gamma}$, performing the expectation in (5.8), and equating the results to zero. These steps are made in appendix A and the following update equations are derived.

$$\tilde{\mathbf{A}}^{(s+1)} = \left(\sum_{n=1}^{N}(\mathbf{x}[n] - \hat{\mathbf{s}}[n])\mathbf{y}^{\mathsf{T}}[n]\right)\left(\sum_{n=1}^{N}\mathbf{y}[n]\mathbf{y}^{\mathsf{T}}[n]\right)^{-1} \tag{5.17}$$

$$\boldsymbol{\Sigma}^{(s+1)} = \frac{1}{N}\sum_{t=1}^{T}\left((\mathbf{x}[n] - \hat{\mathbf{s}}[n] - \tilde{\mathbf{A}}^{(s+1)})(\mathbf{x}[n] - \hat{\mathbf{s}}[n])^{\mathsf{T}} + \mathbf{R}[n]\right). \tag{5.18}$$

Thus, the model parameters can be initialized using (5.5) and (5.6) and then updated using (5.17) and (5.18). The new model parameters will be the old ones for the next iteration. The iterations stop when the increase in the incomplete log-likelihood function is less than some threshold or a maximum number of iterations is exceeded.

Practically, it is desirable to assume special structures for the distortion parameters in order to avoid the curse of dimensionality problem. In addition, the estimation in the testing phase should be done as fast as possible. Therefore, two special structures of the autoregressive and noise covariance matrices are considered:

the diagonal structure and the spherical structure, i.e., diagonal and all elements on the diagonal are the same. One more advantage of these assumptions is its consistency with the practical observation that the noise components are statistically uncorrelated. The update equations for both the diagonal and the spherical structures are obtained in a similar way to that followed in the general case.

For diagonal autoregression matrices and diagonal noise covariance matrices, the update equations are given by

$$\mathbf{a}_d^{(s+1)} = \left(\sum_{n=1}^{N} \mathbf{y}_d[n]\mathbf{y}_d[n]^{\mathtt{T}}\right)^{-1} \left(\sum_{n=1}^{N} (x_d[n] - \hat{s}_d[n])\mathbf{y}_d[n]^{\mathtt{T}}\right) \tag{5.19}$$

$$\sigma_d^{2(s+1)} = \frac{1}{N}\sum_{n=1}^{N} \Big(R_d[n] + (x_d[n] - \hat{s}_d[n])(x_d[n] - \hat{s}_d[n] - (\mathbf{a}_d^{(s+1)})^{\mathtt{T}}\mathbf{y}_d[n])\Big), \tag{5.20}$$

where $x_d[n]$, $s_d[n]$ are the d^{th} component of $\mathbf{x}[n]$ and $\mathbf{s}[n]$, respectively, $\mathbf{a}_d$ is a vector containing the d^{th} elements on the diagonal of $\mathbf{A}_1, \ldots \mathbf{A}_P$, and $\mathbf{y}_d[n]$ is vector containing the d^{th} elements of $\mathbf{x}[n-1], \ldots, \mathbf{x}[n-P]$, in order. The smoothed estimates $\hat{s}_d[n]$ and $R_d[n]$ are defined as $\hat{s}_d[n] \equiv E\left\{s_d[n]|\mathbf{x}[1:N], \theta^{(s)}\right\}$ and $R_d[n] = E\left\{s_d^2[n]|\mathbf{x}[1:N], \theta^{(s)}\right\} - \hat{s}_d^2[n]$ and calculated using (5.15) and (5.16).

For spherical autoregression matrices and spherical noise covariance matrices, define $\mathbf{a} = \begin{bmatrix} a_1 & \ldots & a_P \end{bmatrix}$, where $\mathbf{A}_p = a_p\mathbf{I}$. The update equations are

$$\mathbf{a}^{(s+1)} = \left(\sum_{n=1}^{N} \mathbf{Y}[n]^{\mathtt{T}}\mathbf{Y}[n]\right)^{-1} \left(\sum_{n=1}^{N} \mathbf{Y}[n]^{\mathtt{T}}(\mathbf{x}[n] - \hat{\mathbf{s}}[n])\right) \tag{5.21}$$

$$\sigma^{2(s+1)} = \frac{1}{ND}\sum_{n=1}^{N} \left(\mathrm{tr}(\mathbf{R}[n] + (\mathbf{x}[n] - \hat{\mathbf{s}}[n])^{\mathtt{T}}(\mathbf{x}[n] - \hat{\mathbf{s}}[n] - \mathbf{Y}[n]\mathbf{a}^{(s+1)})\right) \tag{5.22}$$

where $\mathbf{Y}[n] = \begin{bmatrix} \mathbf{x}[n-1] & \ldots & \mathbf{x}[n-P] \end{bmatrix}$.

When additive white noise is present, it is evident that the distortion alters the centers of the distribution of the clean signal as well. In this case, the centers $\boldsymbol{\mu}_m$, $m = 1, \ldots, M$, have to be updated. For deriving the update equation of $\boldsymbol{\mu}_m$, the expectation of the second term in (5.7) should be evaluated since it is a function

in $\boldsymbol{\mu}_m$. Regardless of the type of the noise covariance matrix, it is straightforward to show that the update equation for the centers $\boldsymbol{\mu}_m$ takes the form.

$$\boldsymbol{\mu}_m^{(s+1)} = \frac{\sum_{n=1}^{N} P_{m,n}(\theta^{(s)})\hat{\mathbf{s}}_m[n]}{\sum_{n=1}^{N} P_{m,n}(\theta^{(s)})}, \tag{5.23}$$

5.3 Selection of the optimum regression order

In the above section, we outlined a procedure for estimating the proposed adaptation model parameters given that the regression order, P, is known. However, this is not the case in practical applications. Therefore, a fast and an accurate model order selection criterion is necessary.

Basically, the model order selection algorithm, proposed in section 4.3, is applied to our GML adaptation as shown in Algorithm 5.1 (the number of the Gaussian components is not estimated). In this context, the information criteria are functions in the regression order only. The AIC, the KIC, and the BIC are given by

$$AIC(P) = -2\log \mathrm{p}(X|\theta(P)) + 2|\theta(P)| \tag{5.24}$$

$$KIC(P) = -2\log \mathrm{p}(X|\theta(P)) + 3|\theta(P)| \tag{5.25}$$

$$BIC(P) = -2\log \mathrm{p}(X|\theta(P)) + |\theta(P)|\log N \tag{5.26}$$

For convolutive noise, the number of parameters is given by

$$|\theta(P)| = \begin{cases} D^2(P+1) & \text{general structure} \\ D(P+1) & \text{diagonal structure} \\ P+1 & \text{spherical structure} \end{cases}. \tag{5.27}$$

For additive noise, a constant term MD should be added to $|\theta(P)|$. Of course, this method results in a great saving in time but it heavily depends on the proper estimation of the VARGM model with the least order.

Alternatively, we may replace the incomplete likelihood function with that obtained by ordinary VAR modeling, i.e., assuming there is only one component when

Algorithm 5.1 Selection of the optimal regression order of the GML adaptation algorithm.

1: **Inputs:** X, M, $\mathrm{p}(\mathbf{s}[n])$.

2: **Output:** Optimal P and $\theta(P)$.

3: Estimate the distortion model parameters, $\theta(0)$, from the data using the EM algorithm.

4: Calculate $P(X|\theta(0))$ using (5.7).

5: Calculate $IC(0)$ using (5.24), (5.25), or (5.26).

6: **for** $P = 1$ to P_{max} **do**

7: Set $\theta_0 = \theta(P-1)$ (Copy all the parameters of $\theta(P-1)$ to θ_0).

8: Increment the regression order of θ_0 by 1.

9: In θ_0, put $\mathbf{A}_P = \mathbf{0}_{D\times D}$.

10: Train $\theta(P)$ using the EM algorithm. Take θ_0 as the initial model for the EM algorithm.

11: Calculate $\mathrm{p}(X|\theta(P))$ using (5.7).

12: Calculate $IC(P)$ using (5.24), (5.25), or (5.26).

13: **end for**

14: The selected regression order P is given by

$$P^* = \arg \min_{P=1,\ldots,P_{max}} IC(P).$$

15: Return $\theta(P)$. This model will be used for adaptation and decision.

calculating the likelihood function. Thus, the model selection criteria can be calculated for different orders in a reasonably small time. Let $\mathcal{L}_0(\lambda)$ be the likelihood function corresponding to ordinary VAR modeling. It can be easily proved that

$$\log \mathrm{p}(X|\theta_0) = -\frac{ND}{2}(\log(2\pi)+1) - \frac{N}{2}\log|\hat{\mathbf{\Sigma}}|, \tag{5.28}$$

where

$$\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}[n] - \tilde{\mathbf{A}}_0\mathbf{y}[n] - \hat{\boldsymbol{\mu}})(\mathbf{x}[n] - \tilde{\mathbf{A}}_0\mathbf{y}[n] - \hat{\boldsymbol{\mu}})^{\mathtt{T}}, \tag{5.29}$$

$$\tilde{\mathbf{A}} = \left(\sum_{n=1}^{N}\mathbf{x}[n]\mathbf{y}^{\mathtt{T}}[n]\right)\left(\sum_{n=1}^{N}\mathbf{y}[n]\mathbf{y}^{\mathtt{T}}[n]\right)^{-1}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}[n] - \tilde{\mathbf{A}}_0\mathbf{y}[n])$$

The first term in (5.28) is irrelevant to our model selection and hence it can be dropped. Similarly, only the first term in (5.27) is a function in P. Hence, we have the following approximate expressions for the AIC, KIC, and BIC, respectively.

$$AIC(P) \approx N\log|\hat{\mathbf{\Sigma}}| + 2PD^2 \tag{5.30}$$

$$KIC \approx N\log|\hat{\mathbf{\Sigma}}| + 3PD^2 \tag{5.31}$$

$$BIC \approx N\log|\hat{\mathbf{\Sigma}}| + PD^2\log(N). \tag{5.32}$$

The method is summarized in algorithm 5.2.

5.4 Adaptation using the GML rule

In sections 5.2 and (5.3), we proposed algorithms for estimating the distortion parameters and selecting the optimal regression order assuming that the speaker of the testing utterance is known. In this section, we integrate these algorithms with speaker classification. The GML rule is a multi-class generalization of the binary-class GLRT, successfully applied in adaptive signal detection [12, 3] and

Algorithm 5.2 Approximate model order selection for the GML adaptation algorithm.

1: **Inputs:** X, M, $\mathrm{p}(\mathbf{s}[n])$.

2: **Output:** Optimal P and $\theta(P)$.

3: **for** $P = 1$ to P_{max} **do**

4: Calculate $\hat{\mathbf{\Sigma}}$ using (5.29).

5: Use (5.30), (5.31), or (5.32) to calculate the approximate model order selection criterion for order P.

6: **end for**

7: The selected regression order P is given by

$$P^* = \arg \min_{P=1,\ldots,P_{max}} IC(P).$$

8: Return $\theta(P)$. This model will be used for adaptation and decision.

voiced-unvoiced speech classification [40]. For binary decision problems, the GLRT decision rule takes the form [66],

$$GLRT = \frac{\max_\theta \mathrm{p}\Big(\mathbf{x}[1:N]\Big|\mathcal{H}_0,\theta\Big)}{\max_\theta \mathrm{p}\Big(\mathbf{x}[1:N]\Big|\mathcal{H}_1,\theta\Big)} \gtrless_{\mathcal{H}_1}^{\mathcal{H}_0} \eta, \tag{5.33}$$

where θ refers to the unknown distortion parameters and $\mathcal{H}_i$ refers to one of the two hypotheses. For multiple-hypotheses classification such as the problems in hand, we replace the GLRT by the GML decision, which takes the form

$$\hat{i} = \arg \max_{i=1,2,\ldots,S} \max_\theta \mathrm{p}\Big(\mathbf{x}[1:N]\Big|\mathcal{H}_i,\theta\Big). \tag{5.34}$$

While the maximization with respect to θ reflects the compensation of the distortion effects, the outer maximization corresponds to the ordinary ML decision rule. Thus, our GML-based speaker identification works as follows. Ordinary processing and feature extraction are applied to the testing utterance and a sequence of the feature vectors is obtained. For each candidate speaker, the clean speech model is substituted by his/her model. The distortion parameters are estimated from the testing feature vectors, as discussed in sections 5.2 and 5.3, and the correspond-

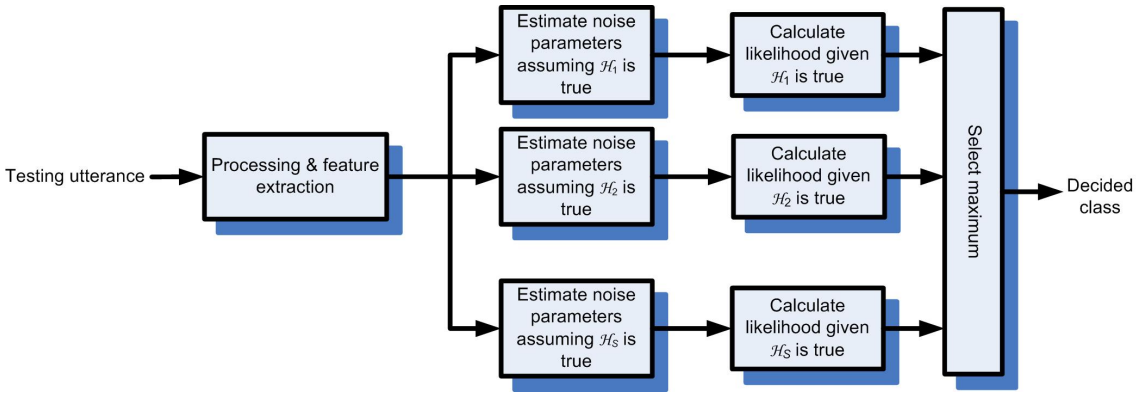

Figure 5.1: The architecture of a GML-based classification system.

ing likelihood value is reported. The decided speaker is the one with the highest likelihood value. An equivalent but more intuitive form of the above rule is

$$\theta_i = \arg\max_{\theta} \mathrm{p}\Big(\mathbf{x}[1:N]\Big|\mathcal{H}_i, \theta\Big)\ ,\ i = 1, ..., S. \tag{5.35}$$

$$\hat{i} = \arg\max_{i=1,2,...,S} \mathrm{p}\Big(\mathbf{x}[1:N]\Big|\mathcal{H}_i, \theta_i\Big) \tag{5.36}$$

An architecture for the proposed compensation technique is depicted in figure 5.1.

5.5 Blind equalization of MIMO channels

In this section, we consider another potential application for the proposed GML decision rule: blind equalization of multiple input multiple output MIMO communication systems. Equalization is defined as the process of restoring a set of source signals distorted by an unknown linear (or nonlinear) filter and possibly an additive noise. Since the 1970's, this problem has been an intensive research topic because it arises in a variety of applications such as speech processing, underwater acoustic, image processing, seismic exploration, and biomedical signal processing. This problem also arises in many digital communication systems such as mobile, wireless communication, sonar, and radar systems.

Depending on the available amount of training data used to estimate the channel impulse response, equalization algorithms can be classified into: non-blind, blind,

and semi-blind equalization algorithms. While non-blind equalization algorithms fully exploit the channel prior information and the available training data to estimate the channel response, blind equalization algorithms attempt to perform the same task without using any training data in order to increase the bandwidth efficiency [112, 109, 46], i.e., to increase the data throughput while preserving low bandwidth consumption. The need for blind equalization is even more critical for channels with frequency selective fading [52]. Blind equalization techniques can be classified into deterministic [118] and statistical. The deterministic techniques are solely based on the subspace decomposition of the received data matrices and, in the absence of noise, they are able to obtain exact estimates within a finite number of observations. Therefore, it is believed that deterministic techniques perform better than statistically-based method when only few observations are available at the receiver. However, when there is a sufficient number of observations, statistically-based methods are superior since they account for the existing noise to some extent by exploiting the statistical properties of the given observations.

Statistically-based blind equalization algorithms are generally divided into two main categories: those based on second-order statistics (SOS) and those based on higher-order (≥ 3) (HOS) statistics. The main motivation behind using HOS has been that, unlike SOS-based methods, HOS are not blind to the phase of the unknown system [23]. HOS-based methods include the inverse filter criteria (IFC) [16, 10, 21, 22], the super exponential (SE) algorithm [77, 124, 61, 68], the polyspectra-based algorithms [78, 20], and the constant modulus algorithm [117, 116, 11], and many others. On the other hand, many researchers were motivated to use SOS in order to reduce the system complexity. Tong, Xu, and Kailath proposed blind equalization of single input multiple output (SIMO) channels using only SOS of systems output [113]. However, the extension to the MIMO channels is not straightforward as long as the system inputs are temporally white. Hua and Tungait proved the identifiability of an MIMO FIR system using SOS when the system inputs are temporally colored. Meanwhile, SOS-based blind equalization algorithms have been reported for the case of temporally white inputs [2].

In [120], the scalar autoregressive model was proposed to equalize ASK modulated signals when transmitted through a SISO channels. In this section, we generalize their approach to deal with MIMO channels. We extend their method in two ways. First, complex time series are considered instead of real ones. This enables us to deal the baseband representation of modulated signals. In addition, and unlike [120], we consider the estimation of the CSI matrix. Moreover, it will be shown that the proposed method can be used in both equalizing the channel effects and estimating the frequency response of the communication channel.

In particular, we prove that under some reasonable conditions, the MIMO channel can be modeled by our proposed VARGM model. The parameters of the VARGM model are estimated from the received symbols using the EM algorithm [29]. The estimated VARGM filter is then used to *equalize* the communication channel. A Bayesian decision rule is applied to the filter output in order to decide about the transmitted symbols. The proposed technique depends only on SOS and hence, it is easy to implement. Moreover, the proposed algorithm requires no prior knowledge of neither the channel response nor the SNR at the receiver. It should be mentioned that the EM algorithm was used before with nonlinearly modulated signals [81, 65, 24] and recently for linearly modulated SISO channels [82]. In all these papers, the received symbols were modeled by an HMM. Typically, the expectation step is performed using either the forward-backward algorithm [65, 24] or the Viterbi-decoding algorithm [81, 82]. That is, in each iteration of the EM algorithm, the expectation step requires a search among many state sequences. This may be time consuming for many practical applications. On the other hand, the parameter estimation of the VARGM model is much faster since, in each iteration, the parameters are just some statistics of the observed symbols. Moreover, the likelihood function calculation is much simpler and hence fast model selection criteria are implemented in our proposed system.

This section is organized as follows. In subsection 5.5.1, the blind equalization problem is formulated. Sufficient conditions for the validity of modeling channels

by a VARGM model are also given in this subsection. In addition, we clarify the similarity between the MIMO equalization problem and the proposed GML adaptation technique. In subsection 5.5.2, we shall show how to estimate the VARGM model parameters using the EM. The model order selection algorithm is very similar in principle to that explained in section 5.3, and hence, we omit it. Finally, the proposed equalization algorithm is explained in details in subsection 5.5.3.

5.5.1 Problem formulation

Consider a MIMO communication channel with N_T transmitters and N_R receivers. In this thesis, we consider only channels that suffer from slow fading. Therefore, it is reasonable to assume that the channel response does not change significantly during the transmission of a single block of symbols. The complex baseband representation of a MIMO channel is usually modeled by the following relation [114].

$$\mathbf{x}[n] = \sum_{i=0}^{Q} \mathbf{H}_i \mathbf{s}[n-i] + \boldsymbol{\epsilon}[n], \tag{5.37}$$

where $\mathbf{s}[n]$ is an $N_T \times 1$ complex baseband vector representation of the transmitted signals at time n, $\mathbf{x}[n]$ is an $N_R \times 1$ complex baseband vector representation of the received signals at time n, and $\epsilon[n]$ is an $N_R \times 1$ baseband vector representation of the additive white Gaussian noise (AWGN) at time n. The matrices $\mathbf{H}_i, i = 1, 2, .., Q$ represent the CSI. Each noise vector follows the complex Gaussian distribution with a zero mean vector and an arbitrary covariance matrix $\tilde{\boldsymbol{\Sigma}}$, which is sometimes assumed to be diagonal. The noise random vectors are assumed to be independent and identically distributed. The equalization of the above MIMO channel is defined as estimating the transmitted sequence $\mathbf{s}[n]$, $n = 1, 2, ..., N$ given some noisy sequence $\mathbf{x}[n]$, $n = 1, 2, ..., N$ observed at the receivers.

The image of (5.37) in the $z-$domain is

$$\begin{aligned} X(z) &= \left(\sum_{i=0}^{Q} \mathbf{H}_i z^{-i} \right) S(z) + \varepsilon(z) \\ &= H(z)S(z) + \varepsilon(z), \end{aligned} \tag{5.38}$$

where $X(z), S(z)$, and $\varepsilon(z)$ are the $z-$transforms of $\mathbf{x}[n], \mathbf{s}[n]$ and $\epsilon[n]$, respectively. The matrix $H(z)$ can be interpreted as the transfer function of the MIMO channel filter. Equation (5.37) takes the form of a vector moving average (VMA) model. In this thesis, we propose inverting this model to an appropriate VAR model because it is easier to identify and equalize a MIMO channel when it is characterized by a VAR model.

In the following theorem we shall show sufficient conditions under which this inversion is possible.

Theorem 1 *If the channel transfer matrix $H(z)$ can be expressed as the product of a full rank square matrix $C^{-1}(z)$, where $C(z) = \sum_{i=1}^{n_c} \mathbf{C}_i z^{-i}$ is of size $N_R \times N_R$ with $|C(z)| \neq 0 \forall |z| > 1$and an irreducible full column rank rectangular matrix $B(z) = \sum_{i=1}^{n_b} \mathbf{B}_i z^{-i}$ of size $N_R \times N_T$ and if $N_R > N_T$, then there exists at least one finite-degree transfer matrix $A(z) = \mathbf{I}_{N_R} - \sum_{i=1}^{P} \mathbf{A}_i z^{-i}$, where $\mathbf{I}_{N_R}$ is the identity matrix of size $N_R \times N_R$ such that*

$$A(z)H(z) = \mathbf{H}_0 = H(\infty), \forall z \tag{5.39}$$

and

$$2n_c + (N_T + 1)n_b \leq P < \infty \tag{5.40}$$

The proof of this theorem is given in appendix B. The matrix $A(z)$ can be regraded as the MIMO channel equalizer. The above conditions are satisfied for most practical systems [122, 46].

Thus, if the channel transfer matrix, $H(z)$, satisfies the conditions in Theorem 1, the given system in (5.37) can be inverted by simply multiplying $A(z)$ from the left for both sides of (5.38), yielding

$$A(z)X(z) = A(z)H(z)A(z) + A(z)\varepsilon(z),$$

or

$$\mathbf{x}[n] = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{x}[n-1] + \mathbf{H}_0 \mathbf{s}[n] + \mathbf{e}[n], \tag{5.41}$$

where

$$\mathbf{e}[n] = \boldsymbol{\epsilon}[n] - \sum_{i=1}^{P} \mathbf{A}_i \boldsymbol{\epsilon}[n-1].$$

Note that the vectors $\mathbf{e}[n]$, $n = 1, 2, ..., N$ are identically distributed (but possibly dependent); each follows the complex Gaussian distribution with zero mean vector and arbitrary covariance matrix $\boldsymbol{\Sigma}$, which is a function of $\tilde{\boldsymbol{\Sigma}}$ and $\mathbf{A}_i, i = 1, 2, ..., P$. Denoting the possible transmitted symbols by $\mathbf{s}_1$, $\mathbf{s}_2$, ..., $\mathbf{s}_M$, the distribution of the sequence $\mathbf{H}_0\mathbf{s}[n] + \mathbf{e}[n]$ is a complex Gaussian mixture model. The number of Gaussian components is the size of the symbol set, M and the covariance matrices of all components are the same. Moreover, the centers of the Gaussian components are $\boldsymbol{\mu}_m = \mathbf{H}_0\mathbf{s}_m$, $m = 1, 2, ..., M$.[1] The transmitted symbols depend on the modulation scheme used but always known to the receiver in advance, and hence, they are treated as constants. The VARGM model parameters can be collectively represented by the string

$$\lambda = \{w_1, \ldots, w_M, \boldsymbol{\Sigma}, \mathbf{A}_1, \ldots, \mathbf{A}_P, \mathbf{H}_0\}.$$

Thus, upon the conditions mentioned above, the equalization problem can be reformulated as follows. Given some received symbols that are modeled by the relation

$$\mathbf{x}[n] = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{x}[n-i] + \mathbf{H}_0\mathbf{s}[n] + \mathbf{e}[n], \tag{5.42}$$

where $\mathbf{e}[n]$ is a zero-mean complex Gaussian random vector, find the ML-estimate of the transmitted symbols $\mathbf{s}[n]$.

Comparing the equalization model (5.42) to the adaptation model (5.1) in section 5.1, we notice the analogy between the two models. The transmitted symbol vector, $\mathbf{s}[n]$, corresponds to the feature vector of the clean speech while the received symbol vector, $\mathbf{x}[n]$, corresponds to the feature vector of the corrupted speech. To equalize the MIMO channel, we need to determine the best regression order, P, and estimate the equalizer filter, $A(z)$, and the noise covariance matrix, $\boldsymbol{\Sigma}$ as we

[1] Actually, in most practical applications all symbols are equally likely to be transmitted. However, in this thesis, we prefer to consider a more general framework.

did before in the GML adaptation problem. Furthermore, the distribution of $\mathbf{s}[n]$ is completely known prior to adaptation in (5.1) or equalization in (5.42). However, there are some differences between the estimation problems. First, the equalization system is not square, i.e., the number of outputs (receivers) in (5.42) is not equal to the number of inputs (transmitters). Second, there is a channel matrix, $\mathbf{H}_0$, which has to be estimated for equalization. Furthermore, the transmitted vector, $\mathbf{s}[n]$, follows a discrete distribution rather the GMM distribution in the GML adaptation case[2]. In addition, we have to consider complex random variables rather than real ones. Nonetheless, and despite these differences, the estimation algorithm of the equalizer filter parameters is conceptually analogous to that followed in section 5.2 as we shall see shortly.

In this thesis, we propose a four-step procedure for solving this problem. First, the received signal, $\mathbf{x}[n]$, is fitted into (5.42) using the EM algorithm. Second, the channel equalizer $A(z)$ is constructed and used to filter the received signal, $\mathbf{x}[n]$. The Bayesian decision rule is then applied to the filter output in order to determine the most-likely transmitted symbols. Finally, a simple algorithm is proposed and applied to resolve possible permutation and phase ambiguities in the final equalizer output.

5.5.2 Parameter estimation of the equalizer filter

Given some observed sequence $\mathbf{x}[1:N]$, it is required to find the maximum likelihood estimates of the model parameters. The likelihood function of the observed sequence is given by

$$\begin{aligned} \mathrm{p}(\mathbf{x}[1:N]|\lambda) &= \prod_{n=1}^{N} \mathrm{p}(\mathbf{x}[n]|\mathbf{x}[1:n-1],\lambda) \\ &= \prod_{n=1}^{N} \left(\sum_{m=1}^{M} w_m \mathcal{CN}(\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n]; \mathbf{H}_0\mathbf{s}_m, \boldsymbol{\Sigma}) \right), \end{aligned} \tag{5.43}$$

[2]In fact, the distribution of $\mathbf{s}[n]$ can be also considered as a GMM with zero covariance matrices.

where

$$\tilde{\mathbf{A}} \equiv \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & ... & \mathbf{A}_P \end{bmatrix},$$

$$\mathbf{y}[n] \equiv \begin{bmatrix} \mathbf{x}^{\mathtt{T}}[n-1] & \mathbf{x}^{\mathtt{T}}[n-2] & ... & \mathbf{x}^{\mathtt{T}}[n-P] \end{bmatrix}^{\mathtt{T}},$$

and $\mathcal{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ refers to the *complex* Gaussian distribution with (complex) mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, given by

$$\mathcal{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\pi^D |\boldsymbol{\Sigma}|} \exp\left(-(\mathbf{x}-\boldsymbol{\mu})^{\dagger} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right), \tag{5.44}$$

where † denotes conjugate transpose. In (5.43), it is assumed that $\mathbf{x}[n] = \mathbf{0}$ whenever $n < 0$. Since the likelihood is nonlinear in the model parameters, the EM algorithm is used for estimating the VARGM model parameters in the ML sense.

For the problem in hand, the complete data specification is similar to that used with the VARGM model, i.e., $Z = \{\Phi, X\}$, where X is the set of received vectors, $\Phi = \{\phi[1:N]\}$ and $\phi[n]$ is the index of the symbol selected at time n. It is straightforward to show that the auxiliary function for our proposed VARGM model is given by

$$\mathrm{Q}(\lambda; \lambda^{(s)}, X) = -N_R N \log(\pi) - N \log|\boldsymbol{\Sigma}| + \sum_{m,n} P_{n,m}(\lambda^{(s)}) \left(\log w_m - \mathbf{e}_m[n]^{\dagger} \boldsymbol{\Sigma}^{-1} \mathbf{e}_m[n]\right), \tag{5.45}$$

where $\sum_{m,n}$ is a short hand for $\sum_{m=1}^{M} \sum_{n=1}^{N}$,

$$\mathbf{e}_m[n] = \mathbf{x}[n] - \sum_{i=1}^{P} \mathbf{A}_i \mathbf{x}[n-i] - \mathbf{H}_0 \mathbf{s}_m,$$

$$P_{n,m}(\lambda^{(s)}) = \frac{w_m^{(s)} \mathcal{CN}(\mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)} \mathbf{y}[n]; \mathbf{H}_0^{(s)} \mathbf{s}_m, \boldsymbol{\Sigma})}{\sum_{m'=1}^{M} w_{m'}^{(s)} \mathcal{CN}(\mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)} \mathbf{y}[n]; \mathbf{H}_0^{(s)} \mathbf{s}_{m'})}. \tag{5.46}$$

Since $\mathrm{Q}(\lambda; \lambda^{(s)}, X)$ is a real function in complex variables $\mathbf{H}_0$ and $\tilde{\mathbf{A}}$, it is more convenient to employ the Wirtinger calculus [81, 55] for optimizing $\mathrm{Q}(\lambda; \lambda^{(s)}, X)$. For a complex vector $\mathbf{z}$, the differential operators $\frac{\partial}{\partial \mathbf{z}}$ (where $\mathbf{z}^*$ are treated as constant) and $\frac{\partial}{\partial \mathbf{z}^*}$ (where $\mathbf{z}$ are treated as constant) yield the same result obtained by separate differentiation with respect to the real and the imaginary part of the function. Wirtinger proved that the complex differential operators are given by

$$\frac{\partial}{\partial \mathbf{z}} \equiv \frac{1}{2} \left(\frac{\partial}{\partial \Re\mathbf{z}} - \jmath \frac{\partial}{\partial \Im\mathbf{z}} \right),$$

$$\frac{\partial}{\partial \mathbf{z}*} \equiv \frac{1}{2}\left(\frac{\partial}{\partial \Re\mathbf{z}} + \jmath\frac{\partial}{\partial \Im\mathbf{z}}\right),$$

where the $\Re(.)$ and $\Im(.)$ operators extract the real and the imaginary parts of a quantity, respectively. Differentiating $\mathrm{Q}(\lambda;\lambda^{(s)},X)$ with respect to $\tilde{\mathbf{A}}^*$ and $\mathbf{H}_0^*$ and equating the results to zeros, the updated values of $\tilde{\mathbf{A}}$ and $\mathbf{H}_0$ are given by solving the following two linear equations.

$$\sum_{m,n} P_{n,m}(\lambda^{(s)})\Big(\mathbf{x}[n]-\tilde{\mathbf{A}}^{(s+1)}\mathbf{y}[n]-\mathbf{H}_0^{(s+1)}\mathbf{s}_m\Big)\,\mathbf{y}^\dagger[n] = 0, \tag{5.47}$$

$$\sum_{m,n} P_{n,m}(\lambda^{(s)})\Big(\mathbf{x}[n]-\tilde{\mathbf{A}}^{(s+1)}\mathbf{y}[n]-\mathbf{H}_0^{(s+1)}\mathbf{s}_m\Big)\,\mathbf{s}_m^\dagger = 0, \tag{5.48}$$

Differentiating $\mathrm{Q}(\lambda;\lambda^{(s)},X)$ with respect to $\mathbf{\Sigma}$ and equating the result to zero, the update equations are given by

$$\mathbf{\Sigma}^{(s+1)} = \frac{\sum_{m,n} P_{n,m}(\lambda^{(s)})\Re\left\{\mathbf{e}_m^{(s)}[n](\mathbf{e}_m^{(s)}[n])^\dagger\right\}}{\sum_{m,n} P_{n,m}(\lambda^{(s)})}, \tag{5.49}$$

In order to optimize $\mathrm{Q}(\lambda;\lambda^{(s)},X)$ with respect to the model priors w_m, $m=1,...,M$, we should consider the constraint that

$$\sum_{m=1}^{M} w_m = 1.$$

Hence, we should differentiate $\mathrm{Q}(\lambda;\lambda^{(s)},X) + \beta\left(\sum_{m=1}^{M} w_m - 1\right)$ with respect to w_m and equate the result to zero.

$$w_m^{(s+1)} = \frac{1}{N}\sum_{n=1}^{N} P_{n,m}(\lambda^{(s)}), \qquad\qquad m = 1,2,...,M \tag{5.50}$$

One final issue is the choice of a proper initial estimate for the model parameters, $\lambda^{(0)}$. Regarding the model priors and the covariance matrix, it was convenient to assign a constant value, $1/M$, for all priors and an identity matrix for the covariance matrix. A good initial value for the auto-regression matrices, $\mathbf{A}_i, i = 1,2,...,P$ can be simply obtained using the Yule-Walker equations or the Nuttall Strand estimators [76]. Finally, the matrix $\mathbf{H}_0$ is initialized randomly. The estimation procedure is outlined in Algorithm 5.3.

In appendix C, a brief analysis on the convergence of the EM algorithm is given.

Algorithm 5.3 Estimation of the channel equalizer filter using the EM algorithm.

1: **Inputs:** $X = \mathbf{x}[1:N]$, P.

2: **Output:** $\mathbf{H}_0$, $\mathbf{\Sigma}$, and equalizer filter $A(z)$.

3: Set $w_m^{(0)} = 1/M$ for all $m = 1, 2, ..., M$.

4: Set $\Sigma^{(0)} = \mathbf{I}_{N_R}$.

5: Estimate an initial value for $\tilde{\mathbf{A}}$ using the Yule-Walker method.

6: Assign random values for $\mathbf{H}$.

7: $\log \mathrm{p}\big(X|\lambda^{(0)}\big) = -\infty$.

8: **for** $s = 1$ to $s_{\max}$ (max. number of iterations) **do**

9: Calculate the log-likelihood value $\log \mathrm{p}\big(X|\lambda^{(s)}\big)$ using (5.43).

10: If $\log \mathrm{p}\big(X|\lambda^{(s)}\big) - \log \mathrm{p}\big(X|\lambda^{(s-1)}\big) <$ some tolerance, return $\lambda^{(s)}$, otherwise remain in the loop.

11: Calculate $P_{n,m}(\lambda^{(s)})$ for $n = 1, ..., N$ and $m = 1, ..., M$ using (5.46).

12: Calculate $\mathbf{\Sigma}^{(s+1)}$ using (5.49).

13: Calculate $w_m^{(s+1)}$ for $m = 1, ..., M$ using (5.50).

14: Solve (5.47) and (5.48) in order to obtain $\tilde{\mathbf{A}}^{(s+1)}$ and $\mathbf{H}^{(s+1)}$.

15: **end for**

5.5.3 The proposed equalization algorithm

The VARGM model can be used to equalize the MIMO channel as follows. Given a set of observed symbols $\{\mathbf{x}[1], ..., \mathbf{x}[n]\}$, the EM algorithm is used to estimate the VARGM model parameters λ. Define the residual vectors, $\mathbf{w}[n]$, as

$$\begin{aligned}\hat{\mathbf{w}}[n] &\equiv \mathbf{x}[n] - \sum_{i=1}^{P} \hat{\mathbf{A}}_i \mathbf{x}[n-i] \\ &= \hat{\mathbf{H}}_0 \mathbf{s}[n] + \mathbf{e}[n], \end{aligned} \tag{5.51}$$

where $\hat{}$ denotes estimated values and the second line in the above equation is derived from (5.41). Since an estimate of the $\mathbf{H}_0$ is available, one can estimate the transmitted sequence as $\mathbf{H}_0^{\sim -1}\mathbf{s}[n]$, where $\mathbf{H}_0^{\sim -1}$ is the left pseudo-inverse of $\mathbf{H}_0$. However, in order to exploit the noise statistical properties, a Bayesian decision rule may be preferable. At each time instant n, the equalization problem can be formulated as the following M-ary hypothesis testing problem:

$\mathcal{H}_m$: Symbol $\mathbf{s}_m$ was transmitted at instant n.

Given that the true transmitted symbol at time n is $\mathbf{s}_m$, the conditional distribution of $\hat{\mathbf{w}}[n]$ is a complex Gaussian distribution with mean $\mathbf{H}_0\mathbf{s}_m$ and covariance matrix $\mathbf{\Sigma}$. Hence, the index of the decoded symbol at time n, $\hat{\phi}[n]$, can be given by the following Bayesian decision rule.

$$\begin{aligned}\hat{\phi}[n] &= \arg \max_{m=1,...,M} P(\mathcal{H}_m | \hat{\mathbf{w}}[n]) \\ &= \arg \max_{m=1,...,M} P(\mathbf{s}[n] = \mathbf{s}_m) P\left(\hat{\mathbf{w}}[n] | \mathbf{s}[n] = \mathbf{s}_m\right) \\ &= \arg \min_{m=1,...,M} \Big(-\log(\hat{w}_m) + (\hat{\mathbf{w}}[n] - \hat{\mathbf{H}}_0\mathbf{s}_m)^{\dagger} \mathbf{\Sigma}^{-1} (\hat{\mathbf{w}}[n] - \hat{\mathbf{H}}_0\mathbf{s}_m) \Big). \end{aligned} \tag{5.52}$$

Similar to most blind equalization algorithms of MIMO channels, the recovered sequence is identifiable up to phase and permutation ambiguities [114]. Several techniques have been proposed for ambiguity resolution (See [46] and the references therein). In this context, the following short training sequence is sent before

transmitting the actual data

$$\begin{bmatrix} s_1 & s_2 & s_1 & \dots & s_1 \\ s_1 & s_1 & s_2 & \dots & s_1 \\ \vdots & \vdots & \vdots & & \\ s_1 & s_1 & s_1 & \dots & s_2 \end{bmatrix}_{N_T \times (N_T+1)},$$

where s_1 and s_2 are any two possible symbols the transmitter can send. Assuming error free transmission, the permutation ambiguity is resolved by rearranging the rows of the received sequence so that the recovered symbols corresponding to the training sequence will have the following form

$$\begin{bmatrix} r_1 & r_2 & r_1 & \dots & r_1 \\ r_1 & r_1 & r_2 & \dots & r_1 \\ \vdots & \vdots & \vdots & & \\ r_1 & r_1 & r_1 & \dots & r_2 \end{bmatrix}_{N_T \times (N_T+1)}.$$

The phase ambiguity is resolved simply by the comparing r_1 to s_1 and r_2 to s_2. Since the transmission is not noise free, the above training sequence should be sent several (odd number of) times and a majority vote is taken among the received symbols corresponding to each of s_1 and s_2 so as to decide the most-likely transmitted symbol. A functional block diagram of the proposed equalization algorithm (with ambiguity resolving) is depicted in figure 5.2.

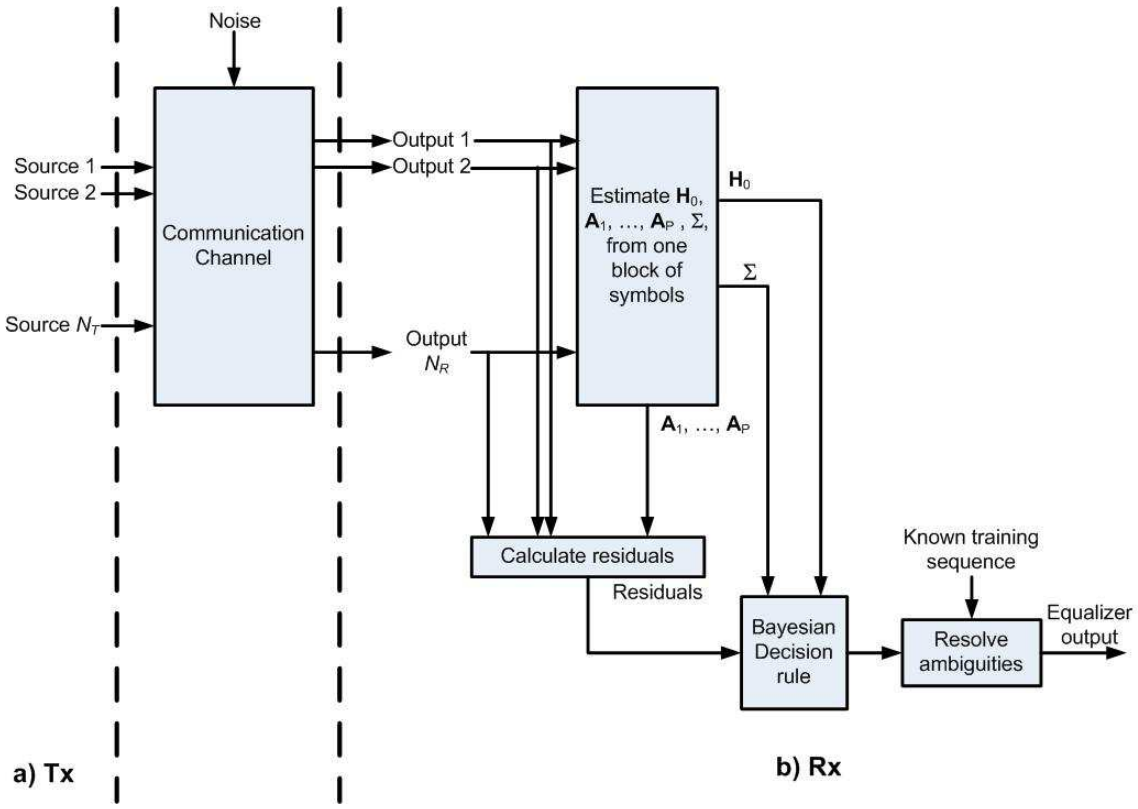

Figure 5.2: Functional block diagram of the proposed equalization algorithm.

Chapter 6

Experimental Evaluation

In this thesis, we performed the following four groups of simulation.

1. We investigated the classification performance of the proposed VARGM classifier for closed-set text-independent speaker identification. We also established a comparison between the VARGM and the GMM classifiers. We used the 2000 NIST speaker recognition evaluation [99] for evaluating the performance.

2. The proposed VARGM was also applied to the speech emotion classification problem. The Berlin emotional database was used in this simulation.

3. We applied the proposed GML adaptation technique to artificially corrupted utterances in the TIMIT database. We examined the performance of our adaptation technique against additive and convolutive noise.

4. The proposed equalization technique, discussed in section 5.5, is applied to three examples and compared against the whitening approach and the BDCC method.

For the first three experiments, the speech processing and feature extraction were almost the same. In order to equalize the effect of the propagation of speech through air, a pre-emphasis radiation filter is used to process speech signal before extraction

of features. In our simulations, we used the following radiation filter

$$H(z) = 1 - 0.97z^{-1}.$$

Hamming windows of duration 25 msec were multiplied by the times samples of each frame. Feature vectors were extracted at a rate of one feature every 10 msec. The MFCC features were extracted from each frame as explained in section 2.2.

6.1 Group I: Closed-set text-independent speaker identification using the VARGM model

We conducted two main simulations to validate the effectiveness of the proposed VARGM classification technique in the closed-set speaker identification problem. In the first experiment, the VARGM was compared to the GMM with fixed model orders for each speaker model. In the second experiment, we studied the effect of our proposed model order selection technique on the classification performance of the system. In all simulations, the maximum number of iterations in the EM estimation algorithm was 100 and the termination tolerance was 5×10^{-7}.

6.1.1 The 2000 NIST speaker recognition evaluation

We used the 2000 NIST speaker recognition evaluation [99] for validating the performance of our system. The 2000 NIST evaluation was mainly developed for four speaker recognition problems: one-speaker detection, two-speaker detection, speaker tracking, and speaker segmentation. Since we are mainly interested on the speaker identification problem, we considered only utterances prepared for the task of the one speaker detection.

The 2000 NIST speaker recognition evaluation consists of 10,328 utterances containing a total of approximately 4.31 Gbytes of data and covering 148.9 hours of audio. All utterances were recorded in a single channel environment with 8-bit/sample mu-law encoding. The sampling rate is 8 kHz. The audio files were

stored in SPHERE format. Utterances were collected from 936 speakers (428 males and 508 females). Most of the speakers uttered one training utterance with an average duration of 2 minutes and from 5 to 28 testing utterances with a duration ranging from 15 to 45 seconds.

All testing utterances were collected from telephone conversations with different dialed numbers and different handset than that used in the speaker's training data. In our simulations, we basically considered utterances recorded from electret handset devices since they constituted the majority of the utterances. The classification accuracy is calculated by simply dividing the number of correctly classified testing utterances over the total number of testing utterances.

6.1.2 A comparison between GMM and VARGM

We compared the performance of our proposed system to that of the standard GMM system. In order to establish reliable conclusions, different numbers of speakers were tried. Moreover, for each number of speakers, we tried three types of populations: all speakers are male, all speakers are female, and half the speakers are male and the other half is female. The mean and the standard deviation of the classification accuracies are shown in Table 6.1. Each entry in Table 6.1 is based on 5 trials. In this experiment, we tried a fixed number of Gaussian components, $M = 128$, for all speaker models and a fixed regression order $P = 3$ for all VARGM speaker models. In this simulation, the combination of M and P is chosen by trial and error. In the simulation in subsection 6.1.3, the performance of the proposed model order selection technique is investigated. Both the autoregression and the noise covariance matrices were assumed diagonal in this simulation.

As expected, with the increase of the number of speakers (classes), the classification task becomes more difficult, and hence, the classification accuracy decreases. Nonetheless, the proposed VARGM model consistently outperforms the standard GMM method for all the configurations. The amount of improvement is between 2% to 5% for most cases. The same experiment was repeated with the incorpora-

Table 6.1: Classification performances of the GMM and the VARGM systems when applied to utterances from the 2000 NIST speaker recognition evaluation.

No. of speakers	Gender	GMM	VARGM
20	Male	$(83.21 \pm 1.40)\%$	$(87.92 \pm 0.79)\%$
	Female	$(85.14 \pm 1.76)\%$	$(91.35 \pm 2.00)\%$
	Mixed	$(85.96 \pm 1.00)\%$	$(87.98 \pm 2.04)\%$
50	Male	$(72.91 \pm 1.19)\%$	$(75.43 \pm 1.92)\%$
	Female	$(74.41 \pm 0.79)\%$	$(75.90 \pm 1.23)\%$
	Mixed	$(74.30 \pm 1.30)\%$	$(75.84 \pm 1.14)\%$
100	Male	$(69.84 \pm 0.69)\%$	$(71.75 \pm 1.18)\%$
	Female	$(67.92 \pm 0.74)\%$	$(70.87 \pm 1.11)\%$
	Mixed	$(67.94 \pm 0.95)\%$	$(70.29 \pm 0.82)\%$

tion of the UBM framework and the classification results are shown in Table 6.2. For the GMM parameters, the update parameters were selected as recommended in [100], i.e., $\nu_m = \tau_m = \alpha_m - d - 1 = 16$ for $m = 1, \ldots, M$. For the auto-regression matrices, the update parameter was selected as

$$\beta = 0.1 \sum_{n=1}^{N} ||\mathbf{y}[n]||^2.$$

In fact, we found experimentally that the classification accuracy was almost insensitive for small values of β. Comparing the classification accuracies of Table 6.1 and 6.2, we observe the improvement achieved by incorporating the UBM model. At the same time, the proposed method still provides improvement in the classification accuracy over the standard GMM system. We should emphasize, however, that both M and P were empirically determined. In fact, for high regression orders, the VARGM classifier may suffer from over-fitting like other classifiers. Therefore, it is important to apply model order selection techniques to adequately determine *good* values for both M and P.

Table 6.2: Classification performances of the GMM-UBM and the VARGM-UBM systems when applied to utterances from the 2000 NIST speaker recognition evaluation.

No. of speakers	Gender	GMM	VARGM
20	Male	$(88.68 \pm 1.16)\%$	$(89.81 \pm 1.81)\%$
	Female	$(82.86 \pm 1.48)\%$	$(84.03 \pm 2.75)\%$
	Mixed	$(79.57 \pm 1.17)\%$	$(83.83 \pm 0.89)\%$
50	Male	$(75.83 \pm 1.26)\%$	$(76.93 \pm 2.08)\%$
	Female	$(78.83 \pm 0.53)\%$	$(79.9 \pm 3.76)\%$
	Mixed	$(77.29 \pm 1.11)\%$	$(79.91 \pm 1.11)\%$
100	Male	$(70.95 \pm 0.84)\%$	$(73.72 \pm 1.67)\%$
	Female	$(69.60 \pm 0.64)\%$	$(71.66 \pm 4.02)\%$
	Mixed	$(70.99 \pm 1.04)\%$	$(74.12 \pm 2.90)\%$

6.1.3 VARGM model order selection

In this simulation, we investigated the effect of model order selection on the classification performance of our proposed method as well the standard GMM system. Algorithms 4.2 and 4.3 ($P = 0$) were used to select the order of the GMM and the VARGM speaker models, respectively. We basically considered a population of 50 speakers with mixed genders in this simulation.

Table 6.3 shows a comparison between the three model order selection techniques with respect to:

1. the classification accuracy of the VARGM model, $\text{acc}_{\text{VARGM}}$,
2. the average number of Gaussian components in the VARGM models, $\hat{M}_{\text{VARGM}}$,
3. the average regression order of the VARGM models, $\hat{P}_{\text{VARGM}}$,
4. the classification accuracy of the GMM models, acc_{GMM},
5. the average number of Gaussian components in the GMM models, $\hat{M}_{\text{GMM}}$,

Table 6.3: Classification performance of the AIC, the BIC, and the KIC model order selection techniques for the 2000 NIST database.

Selection criterion	$\text{acc}_{\text{VARGM}}$	$\hat{M}_{\text{VARGM}}$	$\hat{P}_{\text{VARGM}}$	acc_{GMM}	$\hat{M}_{\text{GMM}}$
AIC	79.25%	98.56	4.08	78.09%	256.00
KIC	75.39%	66.56	3.00	73.71%	225.28
BIC	82.60%	64.00	1.92	78.48%	65.28

It is obvious from Table 6.3 that the VARGM model still outperforms the GMM model by 1% - 4% in the classification accuracy. Comparing the classification accuracies in Table 6.3 to the corresponding accuracies in Table 6.1, a significant improvement in the accuracy is observed specially with the BIC. This indicates the importance of applying model order selection technique for increasing the classification accuracy. Comparing the three model selection criteria, we find that the BIC provides the highest classification accuracy and the simplest classifiers. This advantage in performance may be attributed to the fact that the BIC accounts for the number of data points, unlike the other two criteria. According to the literature of pattern classification, it is argued that, to some extent, simpler classifiers have better generalization capabilities [33].

6.2 Group II: Speech emotion recognition using the VARGM model

Another recent application to the proposed VARGM-based classification framework is speech emotion classification [6], which refers to the process of determining the emotional state of a speaker from his voice. Recently, there has been an increasing research interest in speech emotion classification for it has found a variety of applications such as web interactive movies, information retrieval, medical analysis, in-car board systems and text-to-speech synthesis [105].

Many classification techniques have been applied for speech emotion classification such as ANN [59], the HMM [91] and the SVM [105]. However, an important remark in the majority of these techniques is that they do not model the temporal structure of the training data. The only exception is the HMM in which the temporal structure of the data is modeled through its states. However, all the Baum-Welch re-estimation formulae are based on the assumption that, within the same state, all the feature vectors are statistically independent [91]. Though this assumption is not valid in practice, the HMM has shown to be a powerful classifier in a variety of applications.

In this section, we compare the classification performance of the proposed VARGM modeling technique with that of the HMM, the k-NN, and the ANN classification methods. While the k-NN and the ANN classifiers do not model timing dependency altogether, the HMM models timing dependency through state transition. In addition, the HMM is very popular in speech applications and has been applied to the problem of speech emotion recognition [91].

Unlike the speaker identification problem, we used VARGM models with full covariance and full autoregression matrices for classification. The main reason is that the duration of all the utterances was small. As a result, the number of extracted feature vectors was so small that the parameter estimation procedure outlined in section 4.2.1 can be easily implemented.

6.2.1 The Berlin emotional database

The VARGM-based classification technique was applied to the Berlin emotional speech database [15], which contains 800 utterances recorded in German with the following adult-directed emotions: *anger*, *boredom*, *fear*, *happiness*, *sadness*, and *neutral*. Ten professional native German actors (5 female and 5 male) simulated these emotions, producing 10 utterances for each emotion (5 short and 5 longer sentences). The script of the utterances could be used in every-day communication and are interpretable in all applied emotions. All utterances were recorded using 16

bit/sample PCM encoding and a sampling frequency of 16 kHz. The recordings were made using a Sennheiser MKH 40 P 48 microphone and a Tascam DA P1 portable DAT-recorder in an anechoic chamber. The recognizability and the naturalness of the utterances were tested by 20-30 judges. The human recognition rate was more than 80%.

In order not to favor one of the emotions over the others, the number of training and testing utterances should be the same for all emotions. Since the total number of utterances for each emotion is variable, only fifty utterances are randomly selected without replacement from each emotion. At the time of this simulation, the number of utterances for the *disgust* emotion was fairly low and hence this emotion was discarded from the experiments[1]. All the recognition accuracies are estimated based on five-fold cross validation. Therefore, the utterances of each emotion is divided into 5-folds with 10 utterances in each. Each recognition accuracy is the average of 5 recognition accuracies obtained by 5 different runs. In each run, we have 40 training utterances (4 folds) and 10 testing utterances (1 fold) for each emotion. The role of folds used for training and testing is switched in each run.

6.2.2 Results and discussion

In the following simulations, learning and classification of the HMM were implemented using the hidden Markov toolkit (HTK) [126] thanks to its reliable performance. The number of hidden layers in the ANN was fixed to two layers and the back-propagation algorithm is used to train the network.

For all the classification techniques, it was necessary to apply a model selection technique to determine the following structural parameters: the number of neighbors in the kNN classifiers, the number of nodes in each hidden layer of the ANN classifiers, the number of states and the number of Gaussian components per states in the HMM, and the regression order and the number of Gaussian components in

[1] At the time of this simulation, not all the utterance were available. That is the main reason behind using a relatively small number of utterances.

the VARGM model. Since the number of extracted feature vectors per utterance was limited, it was unreliable to use information-theoretic model selection criteria such as the AIC and the BIC. Therefore, the model order selection techniques presented in section 4.3 were not applied in this simulation. Instead, we applied another model selection technique that is based on cross-validation [[13], ch.9]. In particular, for each possible setting of the structural design parameters of the classifier, five-fold cross validation technique was applied to the training data only. The model selection criterion is the average cross validation error. Once the optimal model order is determined, all the training data is used to retrain the selected model and the accuracy with respect to the test set is reported.

In order to demonstrate the importance of modeling the dependency between successive feature vectors, the cross validation accuracy was calculated for different combinations of M and P. The obtained accuracies are then averaged with respect to M and plotted versus P. A plot of the average validation accuracy versus P is shown in figure 6.1. Obviously, the case of $P = 0$ corresponds to a pure Gaussian mixture model, i.e., there is no modeling of correlations between feature vectors. It is noted that the accuracy asymptotically increases in general with the increase of P up to a certain regression order. This corresponds to modeling the correlation between a larger number of successive vectors. Thus, taking such a dependency into account results in an increase in the classification accuracy. However, and similar to many other classifiers, the accuracy asymptotically decreases when P is too large since the model may be over-fitted to the distribution of the training data.

Table 6.4 shows the recognition accuracy, the classification time and the selected structural parameters of all the classification techniques. It is noted from the table that the classification accuracy corresponding to the VARGM classifier is higher than the peak accuracy in Figure 6.1. This is expected since more training data is used to estimate the VARGM parameters. It can also be deduced from the table that techniques that model timing dependency (proposed and HMM) generally outperform other techniques, which completely ignore the temporal profile of the

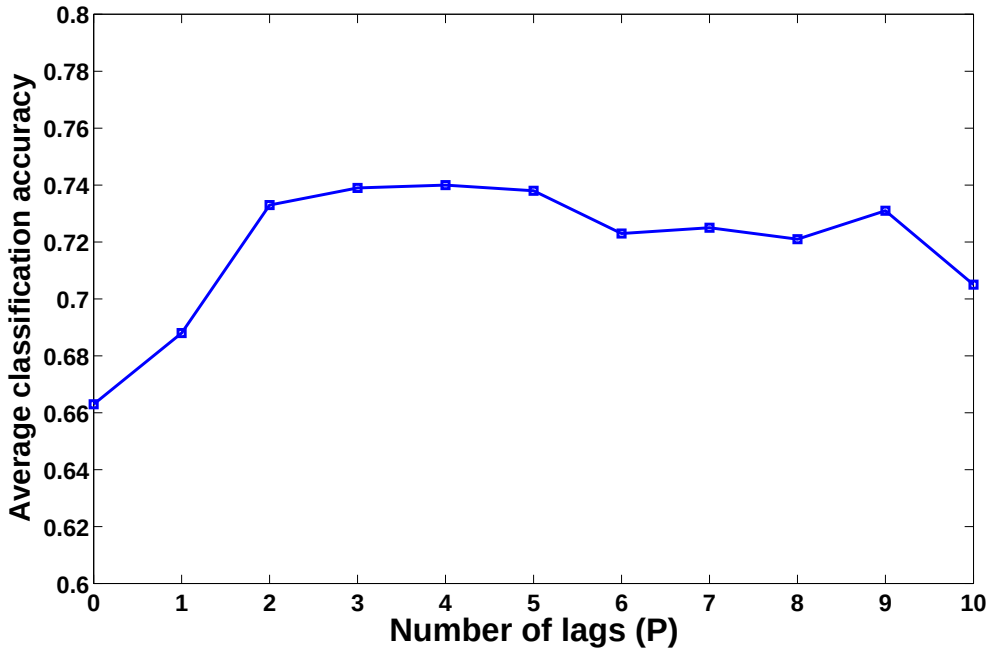

Figure 6.1: Average recognition accuracy of the VARGM recognizer when applied to the Berlin emotional speech database.

sequence of feature vectors. Comparing the identification times of different techniques, it is clear that the average time required by the k-NN is from one to two order of magnitudes higher than other methods. This may be undesirable for many practical applications. On the other hand, the average identification times of other techniques are almost comparable. In addition, the recognition performance of the ANN is inferior to other techniques. According to literature, it seems that ANNs are not well suited for speech emotion recognition [59]. Based on Table 6.4, it may be deduced that the proposed recognition technique achieves the best compromise between the recognition accuracy and the recognition time.

The normalized confusion matrices for both the proposed technique and the HMM technique (the second best recognition method) are shown in Tables 6.5 and 6.6, respectively. Grouping the emotions into three sets: high-arousal emotions (anger, fear, and happiness), low-arousal emotions (boredom and sadness), and the neutral emotion, it is noted that the confusion between two emotions in the same

Table 6.4: Recognition accuracies, average identification times, and selected structural parameters of different recognition techniques when applied to the Berlin emotional speech database.

Classification method	Average Accuracy	Classification time (seconds)	Selected structural parameters
VARGM	76.0%	0.3253	$M = 2$ & $P = 9$
HMM	71.0%	0.3505	$M = 6$ & # states = 5
k-NN	67.3%	16.2132	# neighbors = 6
ANN	55.0%	0.2573	# neurons = 5

Table 6.5: Normalized confusion matrix of the VARGM recognition technique when applied to the Berlin database.

	Recognized emotion					
True emotion	anger	fear	happiness	boredom	sadness	neutral
anger	**0.74**	0.08	0.16	0	0	0.02
fear	0.08	**0.66**	0.12	0	0.04	0.10
happiness	0.18	0.18	**0.62**	0	0	0.02
boredom	0	0.02	0.02	**0.76**	0.04	0.16
sadness	0	0	0	0.02	**0.96**	0.02
neutral	0	0.02	0.04	0.12	0	**0.82**

set is higher than the confusion between two emotions in different sets. This is consistent with what is reported in the literature [91]. From Table 6.5 and 6.6, it can be easily deduced the accuracy of recognition between high-arousal emotions, low-arousal emotions, and the neutral emotion is 90.33% for the proposed method versus 86.00% for the HMM technique. This is intuitive since the speech rate for low-arousal emotions is significantly less than that of high-arousal ones. Hence, there should be a difference in the temporal profile of features extracted from the two emotion categories.

Table 6.6: Normalized confusion matrix of the HMM classifier when applied to the Berlin database.

	Recognized emotion					
True emotion	anger	fear	happiness	boredom	sadness	neutral
anger	**0.78**	0.06	0.16	0	0	0
fear	0.04	**0.7**	0.16	0.04	0.02	0.04
happiness	0.24	0.04	**0.68**	0	0	0.04
boredom	0	0.04	0	**0.42**	0.16	0.38
sadness	0	0	0	0.04	**0.94**	0.02
neutral	0	0.08	0	0.16	0.02	**0.74**

6.3 Group III: Adaptive speaker identification using the GML rule

The robustness of our proposed GML adaptation method was tested by modeling the mismatch between the training and the testing environments by either additive(white) noise or convolutive noise. Basically, clean utterances from the TIMIT database were used to train the speakers' GMMs, while our proposed adaptation technique was applied to artificially corrupted utterances from the same database.

6.3.1 The TIMIT database

The TIMIT database is designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. TIMIT contains broadband recordings of 630 speakers (438 male, 192 female) of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions. All utterances are recorded in a single channel environment with 16-bit/sample PCM encoding. The sampling rate is 16 kHz. All utterances are recorded in noise-free environment.

Though the TIMIT database was mainly designed for evaluating continuous speech recognition systems, it has been used extensively in speaker recognition applications [49, 4, 97]. In our simulations, 8 utterances for each speaker were used for training while the remaining two were used for testing.

6.3.2 Modeling the mismatch by convolutive noise

The effect of the convolutive noise can be modeled as an additive noise in the Mel domain [96]. In the testing phase, the extracted feature vectors are artificially corrupted according to the following model

$$\mathbf{x}[n] = \mathbf{A}_1\mathbf{x}[n-1] + \mathbf{s}[n] + \mathbf{e}[n], \tag{6.1}$$

where $\mathbf{s}[n]$, $\mathbf{e}[n]$, and $\mathbf{s}[n]$ refer to feature vectors extracted from the clean speech (of the TIMIT database), the noise, and the corrupted speech, respectively (see equation (5.1)). The noise vectors are randomly generated according to the multivariate normal distribution with zero mean and diagonal covariance matrix. $\mathbf{A}_1$ is a diagonal matrix with random entries on the diagonal ranging from 0 to 0.5. The matrix $\mathbf{A}_1$ is then scaled to ensure the stability of the system given by (6.1). The noise power is adjusted to fit each desired value of the SNR, given by

$$SNR = \frac{\sum_{n=1}^{N} ||\mathbf{s}[n]||^2}{\sum_{n=1}^{N} ||\mathbf{e}[n]||^2}, \tag{6.2}$$

where N is the number of speech frames in the testing utterance. Note that the reciprocal of (6.2) can be regarded as a measure of the mismatch between the training and the testing environments. Thus, the sequence of the noise feature vectors, $\mathbf{e}[1:N]$, is generated according to the following relation

$$\mathbf{e}[n] = 10^{-SNR/20}\sqrt{\frac{\sum_{n=1}^{N} ||\mathbf{s}[n]||^2}{\sum_{n=1}^{N} ||\mathbf{e}_1[n]||^2}}\mathbf{e}_1[n], \tag{6.3}$$

where the SNR is measured in dB and $\mathbf{e}_1[1:N]$ is a sequence of iid vectors generated according to the standard multivariate distribution $\mathbb{N}(\mathbf{0}, \mathbf{I})$.

In the training phase, the feature vectors of each speaker are fitted into a 3-component GMM (full covariance) using the EM algorithm. The algorithm stops when the increase in the log-likelihood function is less than 5×10^{-7} or the number of iterations exceeds 250. In the testing phase, both the ML and the GML decision rules are used to classify the testing utterances. For the GML rule, the centers $\boldsymbol{\mu}_m$, $m = 1, 2, ..., M$ are assumed unaffected by convolutive noise and thus kept fixed.

Table 6.7 shows a comparison between the classification accuracies of two GMM-based classifiers: system 1 applies ML classification rule and system 2 applies GML classification rule. Only the 50 speakers with the longest recordings are considered in this simulation. The SNR was varied from 0 (strong mismatch between training and testing environments) to 20 dB (negligible mismatch between training and testing environments) with a step of 5 dB. All the classification accuracies are assessed based on 5-fold cross validation, which is repeated 10 times, i.e., each entry in Table 6.7 is based on 50 estimates of the accuracy. Generally, classifier 2 outperforms the standard system for small values of the SNR. For SNRs in the range 0 to 20 dB, the improvement in the classification accuracy is 3%-4% for systems with general covariance matrices, 17%-24% for systems with diagonal covariance matrices, and 44%-70% for systems with spherical covariance matrices. With the increase of the SNR, the difference between the classification accuracies of the two classifiers decreases as expected. For high values of the SNR (15 dB and 20 dB), classifier 1 outperforms the GML system. However, the difference between the recognition accuracies of the standard system and the proposed system with spherical covariance matrices is relatively small because of negligible mismatch between training and testing environments.

It is also noticed that the distortion model with spherical covariance matrices provide the best classification performance because it represents the closest match to the true distortion model.

We also investigated the adaptation performance when the SNR is known. In

Table 6.7: Classification accuracies of classifiers 1 and 2 when mismatch is modeled by convolutive noise. Number of speakers = 50; GM model order = 3.

System		Classifier 1	Classifier 2		
Underlying model		GMM	VARGM		
Classification rule		ML	GML		
Covariance type		-	spherical	diagonal	general
SNR	0	$(5.18 \pm 2.64)\%$	$(74.50 \pm 6.54)\%$	$(22.75 \pm 2.36)\%$	$(9.55 \pm 2.74)\%$
	5	$(35.86 \pm 5.64)\%$	$(79.05 \pm 4.70)\%$	$(59.50 \pm 5.43)\%$	$(38.60 \pm 5.43)\%$
	10	$(78.30 \pm 3.94)\%$	$(82.60 \pm 4.27)\%$	$(67.50 \pm 8.54)\%$	$(59.50 \pm 3.04)\%$
	15	$(85.56 \pm 3.24)\%$	$(85.30 \pm 3.85)\%$	$(68.10 \pm 6.62)\%$	$(78.80 \pm 3.89)\%$
	20	$(86.20 \pm 5.31)\%$	$(85.65 \pm 3.07)\%$	$(68.25 \pm 7.14)\%$	$(78.40 \pm 3.41)\%$

this case, the covariance matrices are initialized as

$$\mathbf{\Gamma}^{(0)} = 10^{-SNR/10} \left(\sum_{n=1}^{N} ||\mathbf{x}[n]||^2 \right) \mathbf{I}$$

and the same experiment is repeated. We used the same seed for the random number generator in order to have a consistent comparison. The classification accuracies are shown in Table 6.8 from which we notice an improvement in the classification accuracy for small SNRs over the corresponding accuracies in Table 6.7 as expected. This indicates the importance of properly initializing the model parameters. However, the amount of improvement is within an acceptable range for most of the cases specially for systems with spherical covariance matrices.

6.3.3 Modeling mismatch by additive white Gaussian noise

In the simulation of this subsection, noise is added to the clean speech signal before feature extraction. That is, the speech time samples of the corrupted speech, x_t, is given by.

$$x_t = s_t + n_t, \qquad t = 1, \ldots, T, \tag{6.4}$$

Table 6.8: Classification accuracies of classifiers 1 and 2 when mismatch is modeled by convolutive noise. The SNR is known in advance. Number of speakers = 50; GM model order = 3.

System		Standard	Proposed		
Underlying model		GMM	VARGM		
Classification rule		ML	GML		
Covariance type		-	spherical	diagonal	general
SNR	0	$(5.18 \pm 2.64)\%$	$(76.98 \pm 5.56)\%$	$(48.36 \pm 6.40)\%$	$(60.80 \pm 7.48)\%$
	5	$(35.86 \pm 5.64)\%$	$(83.72 \pm 4.88)\%$	$(68.34 \pm 6.20)\%$	$(75.56 \pm 5.83)\%$
	10	$(78.30 \pm 3.94)\%$	$(85.84 \pm 4.26)\%$	$(74.80 \pm 4.74)\%$	$(83.36 \pm 5.57)\%$
	15	$(85.56 \pm 3.24)\%$	$(85.30 \pm 4.10)\%$	$(77.72 \pm 4.32)\%$	$(86.00 \pm 4.12)\%$
	20	$(86.24 \pm 3.60)\%$	$(86.82 \pm 4.19)\%$	$(78.44 \pm 3.54)\%$	$(86.60 \pm 3.34)\%$

where s_t is the corresponding clean speech sample and n_t is the corresponding noise sample. The noise samples are generated according to the standard normal distribution. Similar to the previous simulation, the noise power is adjusted to fit each desired of the SNR, given by

$$SNR = \frac{\sum_{t=1}^{T} |s_t|^2}{\sum_{t=1}^{T} |e_t|^2}, \tag{6.5}$$

Note that the reciprocal of the SNR in (6.5) can also be regarded as a measure of the mismatch between the training and the testing environments. In this case, the centers $\boldsymbol{\mu}_m$, $m = 1, 2, ..., M$ will be altered and have to be estimated from the testing utterance together with the noise covariance matrix.

Training is done in a similar way to the previous section. Table 6.9 shows the recognition performance of classifiers 1 and 2. The SNR was varied from 0 dB (strong mismatch) to 30 dB (negligible mismatch) with a step of 5 dB. Classification accuracies are also based on 5-fold cross validation technique. As noticed from the table, classifier 2 provides higher classification accuracies than classifier 1. In some cases, the increase in the classification accuracy is more than 10%. However, for small values of the SNR ratio, the improvement is notably less than that obtained

Table 6.9: Classification accuracies of classifiers 1 and 2 when mismatch is modeled by additive Gaussian white noise. Number of speakers = 50. GM model order = 3.

System		Standard	Proposed		
Underlying model		GMM	VARGM		
Classification rule		ML	GML		
Covariance type		-	spherical	diagonal	general
SNR	0	$(2.20 \pm 0.27)\%$	$(3.2 \pm 1.89)\%$	$(3.2 \pm 1.68)\%$	$(7.2 \pm 0.84)\%$
	5	$(4.80 \pm 0.76)\%$	$(6.20 \pm 1.72)\%$	$(6.3 \pm 1.44)\%$	$(12.2 \pm 4.51)\%$
	10	$(11.70 \pm 3.56)\%$	$(12.40 \pm 3.56)\%$	$(13.2 \pm 3.60)\%$	$(24.1 \pm 5.35)\%$
	15	$(34.70 \pm 2.28)\%$	$(35.20 \pm 2.77)\%$	$(32.10 \pm 1.85)\%$	$(43.80 \pm 6.75)\%$
	20	$(60.60 \pm 4.31)\%$	$(61.10 \pm 3.91)\%$	$(59.00 \pm 4.51)\%$	$(62.50 \pm 8.27)\%$
	25	$(76.10 \pm 3.78)\%$	$(76.00 \pm 3.64)\%$	$(75.60 \pm 4.83)\%$	$(77.60 \pm 4.45)\%$
	30	$(79.70 \pm 3.09)\%$	$(79.70 \pm 3.09)\%$	$(79.90 \pm 3.17)\%$	$(86.60 \pm 4.42)\%$

with convolutive noise. This is expected since the assumed model for distortion does not exactly match with the actual noise corruption process. Note that both the clean speech signal and the noise signal are bandlimited from 300 to 3300 HZ. This leads to even more deviation of the assumed distortion model from the actual distortion process.

6.4 Group IV: Blind equalization of MIMO channels

In order to demonstrate the efficacy of our proposed blind equalization method, three examples are considered in our experimental evaluation. In the first two examples, the proposed method is compared to the whitening method [114] and the BDCC [104], respectively. In the third example, we considered a separable MIMO communication system, i.e., the MIMO communication system can be separated into two or more smaller MIMO systems. Basically, the comparison criteria are the symbol error probability, the bit error probability (BER), and the normalized

mean square error (NMSE) defined by

$$NMSE = \frac{\left|\left|\mathbf{H}_0 - \hat{\mathbf{H}}_0\right|\right|^2}{||\mathbf{H}_0||^2},$$

where $||.||$ denotes the $l-2$ norm of a matrix and $\mathbf{H}_0$ and $\hat{\mathbf{H}}_0$ refer to the true and the estimated value of $\mathbf{H}_0$, respectively. The SNR is measured as

$$SNR = 10 \log \frac{\sum_{n=1}^{N} \left|\left|\sum_{i=0}^{Q} \mathbf{H}_i \mathbf{s}[n-i]\right|\right|^2}{\sum_{n=1}^{N} ||\mathbf{e}[n]||^2}$$

6.4.1 Comparison with the whitening approach

The proposed technique is applied to Example 1 in [114]. In this example, there are two transmitting antennas and three receiving antennas. The communication channel is modeled by

$$\mathbf{x}[n] = \mathbf{H}_0 \mathbf{s}[n] + \mathbf{H}_1 \mathbf{s}[n-1] + \mathbf{e}[n], \tag{6.6}$$

where

$$\mathbf{H}_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \qquad \mathbf{H}_1 = \begin{bmatrix} -0.6 & -0.5 \\ 0 & 0 \\ -1.2 & -1 \end{bmatrix}.$$

According to Theorem 1 in Chapter 5, there exist a channel equalizer filter with degree $P = 3$. Details of the derivation of $A(z)$ if $H(z)$ is known are given in Appendix B. For comparison purposes, we repeated the same setup applied in Example 1 in [114]. The QPSK modulation scheme is used to modulate the transmitted signals. Data blocks of size 500 symbols are used to estimate the channel model parameters. The designed equalizer is then applied to an independent message of size 3000 symbols. Symbol error probabilities are averaged over only 100 Monte-Carlo simulation runs. The order of the VARGM model is set to 3. Figure 6.2 shows a comparison between the proposed method and the whitening method with respect to the symbol error rate for each user. Estimates of symbol error probabilities

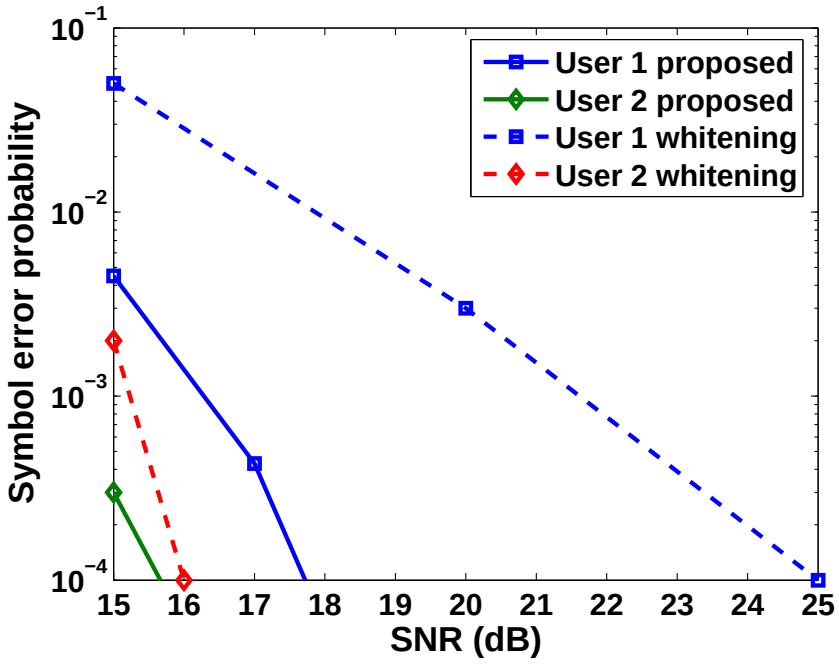

Figure 6.2: Symbol error probability for both the proposed method and the whitening method.

below 10^{-4} are not reliable because of the small number of the Monte-Carlo runs and hence, they are not shown in the figure. From figure 6.2, it is clear that the proposed method consistently outperforms the whitening approach by one order of magnitude over the range of SNRs from 15 to 25 dB.

In order to investigate the performance of different model order selection criteria, data were generated according to (6.6) with $SNR = 15$ dB. In this experiment, we increased the number of Monte-Carlo runs to 1000 to have more reliable estimates of the equalizer parameters. Channel parameters are estimated from the first Monte-Carlo run only and used for the remaining runs. For each run, both the exact and the approximate version of the AIC, the KIC, and the BIC as discussed in section 5.3 are calculated. The selected order according to each criterion is reported. Table 6.10 shows the mean and the standard deviation of the selected orders as well as the overall symbol error probability, and the average equalization time (including time required for parameter estimation and data gathering) for each criterion. Comparing the exact and the approximate versions of model selection

Table 6.10: A comparison between the exact and the approximate versions of the AIC, the KIC, and the BIC.

Model selection criterion	Exact version				Approximate version			
	P_{av}	P_{std}	P_e	t_{eq} (msec)	P_{av}	P_{std}	P_e	t_{eq} (msec)
AIC	8.8540	1.3306	0.0036	6.2489	4.0290	0.6516	0.0038	3.5462
KIC	6.5920	1.3687	0.0035	6.2384	3.2050	0.4349	0.0043	3.3471
BIC	3.9410	0.3971	0.0037	6.2086	1.9990	0.0316	0.0079	3.0792

criteria in terms of the symbol error probability and the equalization time, we conclude Using the AIC and the KIC model order selection techniques the proposed approximation guarantees a relative penalty in the symbol error probability not more than 22.8%. This means that the proposed approximation allows equalization of fluctuations which are 1.76 times faster than using the exact versions of the AIC and KIC model order selection techniques.

6.4.2 Equalization over frequency-flat slow fading channels

The proposed equalization technique is applied to a frequency-flat slow fading channel characterized by

$$\mathbf{x}[n] = \sqrt{\frac{\text{SNR}}{N_T E\left\{|s_1|^2\right\}}}\mathbf{H}_0\mathbf{s}[n] + \mathbf{e}[n], \tag{6.7}$$

where $\mathbf{H}_0$ is a random matrix whose entries are i.i.d. and follow the standard complex Gaussian distribution. For frequency-flat slow fading channels, $\mathbf{H}_0$ can be considered constant during the transmission of a single data block. The factor $E\left\{|s_1|^2\right\}$ represents the average energy of one component of any symbol $\mathbf{s}_m$.

Our proposed technique is compared to the BDCC method [104], which employs low-density parity check (LDPC) encoding for resolving phase and permutation ambiguities. In that paper, a 4×4 block fading channel was simulated. The signals

are modulated using binary PSK modulation scheme. Data blocks of size 100, 400, and 1600 symbols were used to design the equalizers.

In both methods, the NMSE, defined as the relative error in estimating $\mathbf{H}_0$ in decibels, is used as a quality index of the equalization algorithm. Figure 6.3 shows the NMSE of both the proposed and the (2,3)-LDPC-encoded methods. Each point in the curves belonging to the proposed technique is estimated based on 500 Monte-Carlo simulations runs. Except for very small SNR, the proposed technique in general provides less NMSE. Further, the difference between the NMSE of the proposed technique and that of the BDCC technique increases with the increase of the SNR; it reaches about 9 dB when the SNR is 20 dB and 100 symbols are used to estimate the channel equalization filter. A comparison between the BER of the two methods is shown in figure 6.4. The BDCC method generally provides less BER than the proposed method but the difference is not more than 3% for SNR $\geq$ 4dB. For higher SNR, the difference is even much less. It should be mentioned, however, that no error correcting coding scheme was applied. That is, upon the application of our proposed technique, we achieved a great saving in the information rate with the price of a slight increase in the error probability. It is sought that a better detection performance can be obtained if an error correcting coding scheme is incorporated with our proposed equalization technique.

6.4.3 Separable MIMO channels

In some practical situations, the MIMO communication systems can be divided into two (or more) separate MIMO systems. This happens when there is no path between some transmitters and some receivers. It is of interest to us to test the behaviour of our proposed equalization algorithm to detect such situations. In this

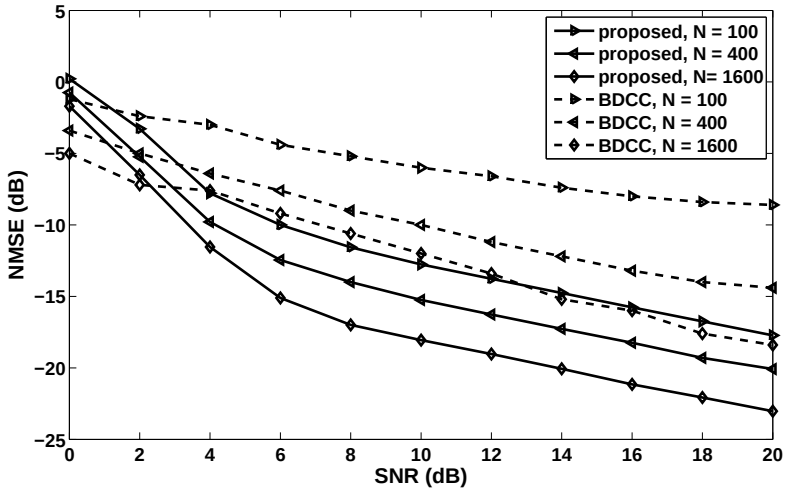

Figure 6.3: NMSE of the proposed method and the BDCC method with block length of 100, 400, 1600 symbols.

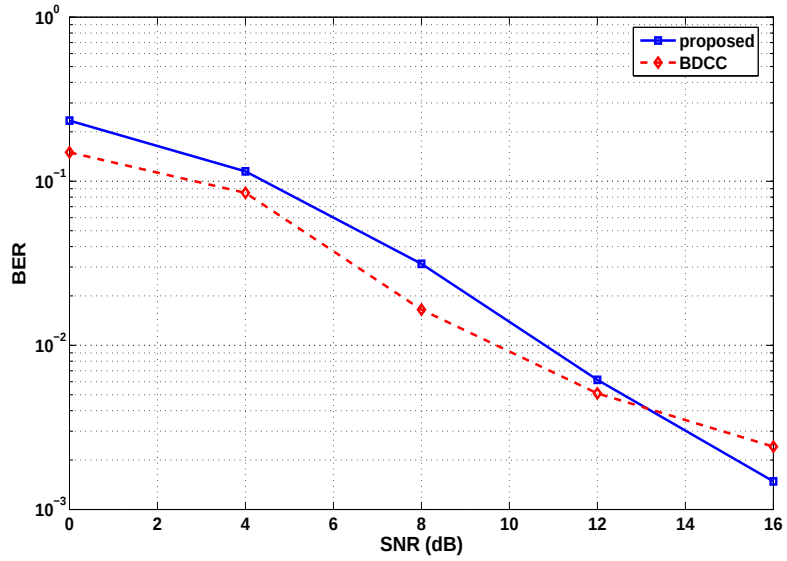

Figure 6.4: BER of the proposed method and the BDCC method with block length of 100 symbols.

example, we simulated the following communication system

$$\mathbf{x}[n] = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0.5 \end{bmatrix} \mathbf{s}[n] + \mathbf{e}[n],$$

Obviously, the zeros in $\mathbf{H}_0$ refer to the nonexistence of a path from a certain transmitter to a certain receiver. Note that the channel matrix $H(z)$ is still irreducible. Data blocks of size 520 symbols are used to estimate the channel model parameters. The designed equalizer is then applied to an independent message of size 10000 symbols. The quadrature phase shift keying (QPSK) modulation scheme is used in this simulation. Based on 5000 Mote-Carlo simulation runs, the symbol error probability is calculated for different values of the SNR ranging from 6 to $16dB$ and compared to the ultimate case in which the channel transfer matrix $\mathbf{H}_0$ is exactly known to the receiver. The comparison is shown in Figure 6.5. The symbol error probability of all the model selection criteria, considered in chapter 5, were almost identical, and hence, only the symbol error probability according to the approximate BIC is plotted. As shown in the figure, the difference between the two error probabilities is less than one order of magnitude for SNRs ranging from 6 to 14 dB. This indicates the accurate estimation of $\mathbf{H}_0$ for a wide range of SNR. This is also evidenced by the small relative error in estimating $\mathbf{H}_0$ shown in figure 6.6. For higher SNRs, the difference increases because the noise covariance becomes smaller and more difficult to estimate.

For each value of the SNR, we measured also the average equalization time per block, t_{eq}. A plot for t_{eq} versus the SNR is depicted in figure 6.7. As shown in the figure, t_{eq} generally decreases with the increase of the SNR. This is consistent with the fact that the lower the SNR the more difficult is to estimate the noise distortion and restore the original transmitted signal. In addition, resolving ambiguities in phase and permutations becomes a harder task for low SNRs.

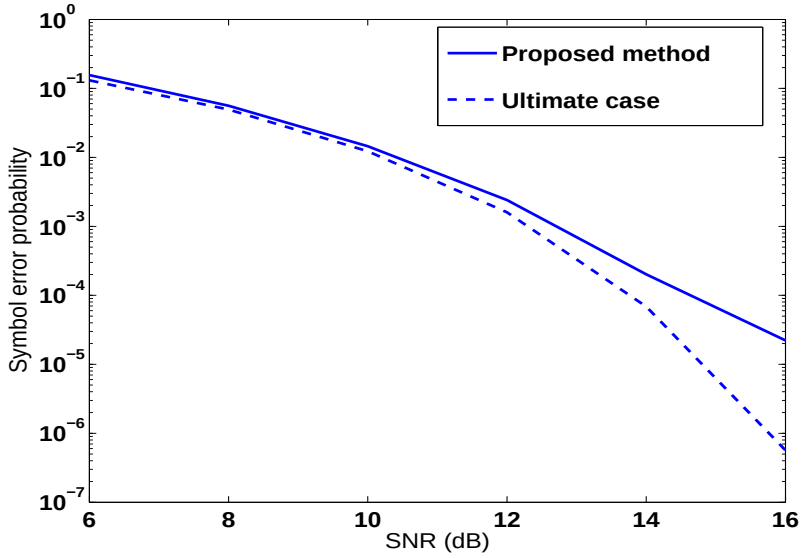

Figure 6.5: A comparison between the symbol error probability of the proposed method and the ultimate equalizer in example 3.

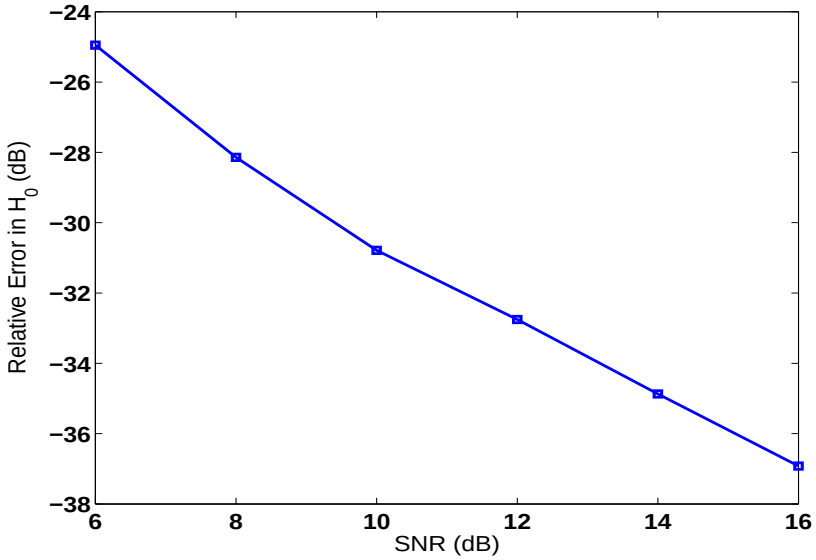

Figure 6.6: Relative error in $\mathbf{H}_0$ in example 3.

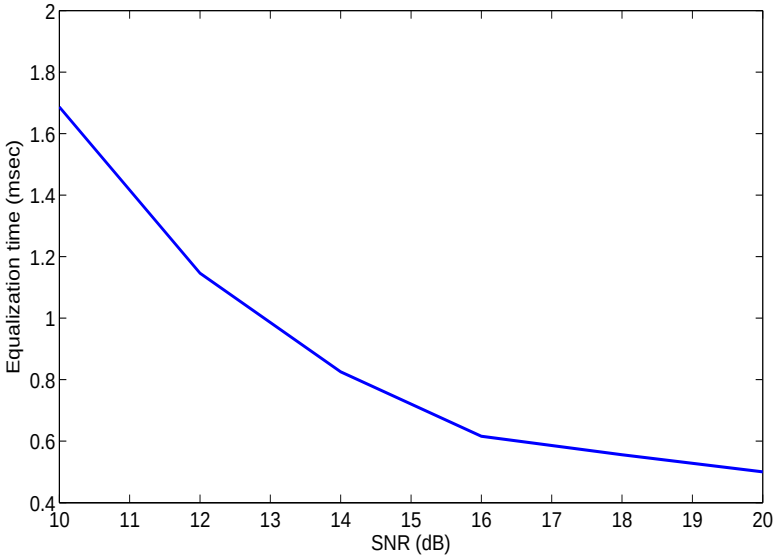

Figure 6.7: Equalization time versus SNR in example 3.

Chapter 7

Conclusions and future work

7.1 Summary of results and thesis contribution

In this thesis, we have primarily investigated the closed-set text-independent speaker identification problem under noisy environment. In particular, we have proposed a two-step procedure for improving the classification performance of the speaker identification system. First, we have proposed a new classifier, the VARGM model, as a combination of the GMM and the VAR models. Thus, the VARGM model has the advantages of modeling the dependency between successive feature vectors and the multi-modality in their distribution. Intuitively, the correlation between feature vectors is caused by extracting features from overlapped frames and the filtering effect of the communication channel through which the speech signal is transmitted. When applied to the 2000 NIST speaker recognition evaluation, the new VARGM classifier has provided 3%-5% improvement in the classification accuracy over the standard GMM classifier.

In the second step in our improvement procedure, we have introduced the GML decision rule as a novel method for compensating the degradation in performance resulting from noise and spectral distortion. The basic idea of the GML adaptation is to assume some parametric form of mismatch between the training and the testing feature vectors. In the testing phase, the testing feature vectors are then used

to estimate these mismatch parameters. To evaluate the efficacy of GML adaptation technique we have modeled the mismatch between the training and testing environments by convolutive noise or additive white Gaussian noise; thus we have applied the GML adaptation technique to utterances from the TIMIT database, artificially corrupted by convolutive and additive white Gaussian noise. The proposed method has shown significant robustness against convolutive noise and notable improvement in accuracy over the standard ML decision rule for utterances corrupted by white noise.

We have also applied the proposed VARGM model to the speech emotion classification problem. The proposed classification technique has been found to provide a better classification performance than other techniques such as the HMM and the kNN in terms of the classification accuracy and the discrimination between high-arousal and low-arousal emotions. This is consistent with the fact that the syllabic rate for low-arousal emotions is significantly less than that of high-arousal ones. Hence, there should be a difference in the temporal profile of features extracted from the two emotion types.

Motivated by the analogy between the GML adaptation technique and the blind equalization problem of MIMO channels, we have proposed a novel technique for the latter problem based on the VARGM modeling. In particular, the received data vectors are fitted into a VARGM model, which is then used to equalize the received data vectors themselves. The most likely transmitted symbols are then determined by applying a fast Bayesian decision rule on the filter output. Finally, permutation and phase ambiguities are resolved using a short training sequence. We have also developed fast procedures for selecting the best regression order of the equalizer filter and estimating its parameters. Compared to other techniques such as the whitening approach and the BDCC method, the proposed method is found to be more accurate in estimating the channel response. In addition, its symbol error probability is less than that of the whitening approach and comparable to that of the LDPC. However, the difference in performance is insignificant with the

advantage that no error correcting code is applied.

7.2 Future research directions

We believe there are many possible extensions to each of the four problems addressed in this thesis. Regarding the speaker identification problem in noisy environments, we have the following suggestions to further improve the classification performance.

- In this work, the VARGM model parameters are estimated using either the ML estimation criterion or the MAP estimation criterion. In the context of speech recognition, discriminative estimation criteria such as the minimum classification error (MCE) [64] and the maximum mutual information (MMI) [8] are found to improve the classification performance of the HMM classifier. Therefore, it is expected that they provide some improvement in the classification performance of our proposed VARGM classifier.

- Based on our study of the speech emotion recognition problem, it seems that the autoregressive part in the VARGM classifier reflects the syllabic rate. In practical applications, it is very likely that the syllabic rate of the training utterance is different from that of the testing utterance for the same speaker. Therefore, we expect more improvement if the autoregression matrices are re-estimated for the testing signal before calculating the likelihood scores. However, the EM algorithm may not be suitable since the estimation of the autoregression should be done as fast as possible in the testing phase.

- We basically assumed the dependency to be in the form of linear regression for mathematical tractability. However, modeling nonlinear correlations may provide us with better characterization of the random process generating the training and testing utterances.

- In the testing phase, the classification is performed after receiving all the feature vectors. For real time applications, it is more practical to redesign our classification algorithm so that the likelihood scores are calculated while accepting feature vectors one by one. Moreover, pruning algorithms such as the nearest neighbor approximation algorithm [87] may be used to speed up the classification process even more.

- Correlation between feature vectors can be modeled in some other ways, which are still mathematically tractable. For example, we may assume the training data modeled as follows.

$$\mathbf{x}[n] = \mathbf{y}[n] + \mathbf{A}_1\mathbf{y}[n-1] + \ldots + \mathbf{A}_p\mathbf{y}[n-P] + \mathbf{e}[n],$$

  where $\mathbf{y}[n]$ follows the GMM distribution and the vectors $\mathbf{y}[1], \ldots, \mathbf{y}[N]$ are iid.

For the proposed GML adaptation technique, the following issues may be considered for future work.

- In the GML adaptation framework, we assumed a particular from for the distribution of the noisy feature vectors. In practical application, it is very difficult to have a general and mathematically tractable form for this distribution. Aggregation of different compensation models can be considered.

- We mainly considered a model-based compensation method. It will be interesting to investigate integrating the proposed method with feature-based and score-based compensation methods.

- In the testing phase, the proposed adaptation technique uses the EM algorithm for estimating the noise parameters. For online or real time applications, the use of iterative algorithms for parameter estimation should be avoided or, at least, minimized.

- In some applications, the number of available testing feature vectors may be so small that the quality of the estimates of the distortion parameters is

affected. This statement was evidenced in section 6.3 by the fact that the best recognition performance occurs when spherical covariance matrices are used. In order to overcome this problem, multiple artificially corrupted versions of the training data can be generated and then used to estimate the distortion parameters.

For speech emotion classification, we observed that the proposed VARGM model classifies well between high-arousal and low-arousal emotions. However, the classification ability between emotions within the same group needs more improvement. Therefore, a possible extension is to study the implementation of a two-stage classifier. In the first stage, emotions are classified into high arousal, low arousal, and neutral emotions using our proposed method. In the second stage, another classifier is used to distinguish between emotions in the same category.

Finally, it will be desirable to investigate the performance of the proposed blind equalization system for MIMO systems with large constellations. In this case, the number of candidate symbols may be so large that equalization cannot be achieved within reasonable time. Recently, Zhao and Davies [129] pointed out this problem and proposed approximating the EM algorithm using the sphere decoding [38] search algorithm. Therefore, a future extension to our equalization method is to incorporate the spherical decoding search algorithm for approximating the summations in the EM algorithms and the maximization in the Bayesian decision rule.

7.3 Publications

7.3.1 Accepted journal papers

1. El Ayadi, M. M., Kamel, M. S., and Karray, F. Toward a tight upper bound for the error probability of the binary Gaussian classification problem. Pattern Recognition. 41, 6 (Jun. 2008).

7.3.2 Submitted journal papers

1. M. El Ayadi, M. Kamel, and F. Karray, Adaptive speaker identification in noisy environment using the generalized maximum likelihood decision rule, to be submitted to IEEE Trans. Audio, Speech, and Language Processing.

2. M. El Ayadi, M. Kamel, and F. Karray, "Modeling and Equalizing Multiple Input Multiple Output Channels using Vector Autoregressive Gaussian Mixture Models", submitted to Signal Processing, Elsevier.

3. M. El Ayadi, M. Kamel, and F. Karray, "Survey on Speech Emotion Recognition: Features, Algorithms, and Applications", submitted to Pattern Recognition, Elsevier.

7.3.3 Accepted conference papers

1. M. El Ayadi, M. Kamel, and F. Karray, Speech Emotion Recognition using Gaussian Mixture Vector Autoregressive Models, IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP 2007, vol. 4, pp. IV-957-IV-960, 2007.

2. M. El Ayadi, M. Kamel, and F. Karray, Time Series Classification using Gaussian Mixture Vector Autoregressive Models, The UW and IEEE Kitchener-Waterloo Section Joint Workshop on Knowledge and Data Mining, 30-31 October, 2006, University of Waterloo, Ontario, Canada.

APPENDICES

Appendix A

Derivation of relations for the smoothed statistics in the GML framework

Fortunately, similar derivations are already established in the context of Kalman filtering [45] where $\mathbf{s}[n]$ corresponds to the hidden states of the Kalman filter and $\mathbf{x}[n]$ corresponds to the observations. The main difference, however, is that the distribution of $\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:N],\theta^{(s)}\Big)$ is not Gaussian but rather it is a mixture of Gaussian, viz,

$$\begin{aligned}\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:N],\theta^{(s)}\Big) &= \mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:n],\theta^{(s)}\Big) \\ &= \sum_{m=1}^{M} \mathrm{p}\Big(\phi[n]=m\Big|\mathbf{x}[1:n],\theta^{(s)}\Big)\,\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:n],\phi[n]=m,\theta^{(s)}\Big)\,.\end{aligned} \tag{A.1}$$

The first line in the above equation is easily derived from the fact that the vectors $\mathbf{x}[n+1:N]$ are conditionally independent of $\mathbf{s}[n]$ given $\mathbf{x}[1:n]$. The *a posteriori*

probability $P_{m,n}(\theta^{(s)}) \equiv \mathrm{p}\Big(\phi[n] = m\Big|\mathbf{x}[1:n], \theta^{(s)}\Big)$ is derived below

$$\begin{aligned} P_{m,n}(\theta^{(s)} &\equiv \mathrm{p}\Big(\phi[n] = m\Big|\mathbf{x}[1:n], \theta^{(s)}\Big) \qquad (A.2)\\ &= \frac{\mathrm{p}\Big(\phi[n] = m\Big|\mathbf{x}[1:n-1], \theta^{(s)}\Big)\,\mathrm{p}\Big(\mathbf{x}[n]\Big|\phi[n] = m, \mathbf{x}[1:n-1], \theta^{(s)}\Big)}{\mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{x}[1:n-1], \theta^{(s)}\Big)} \\ &= \frac{\mathrm{p}(\phi[n] = m)\,\mathrm{p}\Big(\mathbf{x}[n]\Big|\phi[n] = m, \mathbf{x}[1:n-1], \theta^{(s)}\Big)}{\sum_{m'=1}^{M}\mathrm{p}\Big(\phi[n] = m'\Big|\theta^{(s)}\Big)\,\mathrm{p}\Big(\mathbf{x}[n]\Big|\phi[n] = m', \mathbf{x}[n-P:n-1], \theta^{(s)}\Big)} \\ &= \frac{w_m \mathbb{N}(\mathbf{x}[n]; \tilde{\mathbf{A}}\mathbf{w}[n] + \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m + \boldsymbol{\Gamma})}{\sum_{m'=1}^{M} w_{m'}\mathbb{N}(\mathbf{x}[n]; \tilde{\mathbf{A}}\mathbf{w}[n] + \boldsymbol{\mu}_{m'}, \boldsymbol{\Sigma}_{m'} + \boldsymbol{\Gamma})}. \qquad (A.3) \end{aligned}$$

The conditional probability $\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:n], \phi[n] = m, \theta^{(s)}\Big)$ is rewritten as

$$\begin{aligned} &\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:n], \phi[n] = m, \theta^{(s)}\Big) \\ &= \frac{\mathrm{p}\Big(\mathbf{s}[n]\Big|\phi[n] = m, \mathbf{x}[1:n-1], \theta^{(s)}\Big)\,\mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{s}[n], \mathbf{x}[1:n-1], \phi[n] = m, \theta^{(s)}\Big)}{\mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{x}[1:n-1], \phi[n] = m, \theta^{(s)}\Big)} \\ &\propto \mathrm{p}\Big(\mathbf{s}[n]\Big|\phi[n] = m, \theta^{(s)}\Big)\,\mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{s}[n], \mathbf{x}[1:n-1], \phi[n] = m, \theta^{(s)}\Big), \\ &\propto \mathrm{p}\Big(\mathbf{s}[n]\Big|\phi[n] = m, \theta^{(s)}\Big)\,\mathrm{p}\Big(\mathbf{x}[n]\Big|\mathbf{s}[n], \mathbf{x}[1:n-1], \theta^{(s)}\Big), \\ &\propto \mathbb{N}(\mathbf{s}[n]; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\mathbb{N}(\mathbf{x}[n]; \tilde{\mathbf{A}}\mathbf{y}[n] + \mathbf{s}[n], \boldsymbol{\Gamma}), \qquad (A.4) \end{aligned}$$

where the proportionality constant should be chosen so that $\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:n], \phi[n] = m, \theta^{(s)}\Big)$ is a valid density in $\mathbf{s}[n]$. From (A.4), we can easily deduce that $\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:n], \phi[n] = m, \theta^{(s)}\Big)$ is a normal density in $\mathbf{s}[n]$. Hence, it can be expressed in the form

$$\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:n], \phi[n] = m, \theta^{(s)}\Big) = \mathbb{N}(\mathbf{s}[n]; \hat{\mathbf{s}}[n|m], \mathbf{R}[n|m]). \qquad (A.5)$$

Comparing (A.5) with (A.4), we can deduce that

$$\begin{aligned} &(\mathbf{s}[n] - \hat{\mathbf{s}}[n|m])^{\mathsf{T}}\mathbf{R}^{-1}[n|m](\mathbf{s}[n] - \hat{\mathbf{s}}[n|m]) \\ &= (\mathbf{s}[n] - \boldsymbol{\mu}_m^{(s)})^{\mathsf{T}}(\boldsymbol{\Sigma}_m^{(s)})^{-1}(\mathbf{s}[n] - \boldsymbol{\mu}_m^{(s)}) \\ &+ (\mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}[n] - \mathbf{s}[n])^{\mathsf{T}}(\boldsymbol{\Gamma}^{(s)})^{-1}(\mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}[n] - \mathbf{s}[n]) + f, \qquad (A.6) \end{aligned}$$

where f does not depend on $\mathbf{s}[n]$. Equating the quadratic coefficients in $\mathbf{s}[n]$, we get

$$\begin{aligned}\mathbf{R}^{-1}[n|m] &= (\boldsymbol{\Sigma}_m^{(s)})^{-1} +^{\mathrm{T}} (\boldsymbol{\Gamma}^{(s)})^{-1}, \\ \Rrightarrow \mathbf{R}[n|m] &= \boldsymbol{\Sigma}_m - \boldsymbol{\Sigma}_m(\boldsymbol{\Gamma} + \boldsymbol{\Sigma}_m)^{-1}\boldsymbol{\Sigma}_m, \\ &= (\mathbf{I} - \mathbf{K}_m)\boldsymbol{\Sigma}_m, \end{aligned} \tag{A.7}$$

where $\mathbf{K}_m = \boldsymbol{\Sigma}_m(\boldsymbol{\Gamma} + \boldsymbol{\Sigma}_m)^{-1}$. Equating the linear coefficient in $\mathbf{s}[n]$, we have

$$\mathbf{R}^{-1}[n|m]\mathbf{s}[n|m] = (\boldsymbol{\Sigma}_m^{(s)})^{-1}\boldsymbol{\mu}_m^{(s)} +^{\mathrm{T}} (\boldsymbol{\Gamma}^{(s)})^{-1}(\mathbf{x}[n] - \tilde{\mathbf{A}}^{(s)}\mathbf{y}[n]) \tag{A.8}$$

Multiplying (A.8) by $\mathbf{R}^{-1}[n|m]$, and substituting (A.7) into into the resultant equations, we get

$$\mathbf{s}[n|m] = \boldsymbol{\mu}_m + \mathbf{K}_m(\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n] - \boldsymbol{\mu}_m). \tag{A.9}$$

Substituting (A.2) and (A.5) into (A.1), the GMM distribution of $\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:N], \theta^{(s)}\Big)$ is in the form

$$\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:N], \theta^{(s)}\Big) = \sum_{m=1}^{M} P_{m,n}(\theta^{(s)})\mathbb{N}(\mathbf{s}[n]; \hat{\mathbf{s}}[n|m], \mathbf{R}[n|m]), \tag{A.10}$$

where $P_{m,n}(\theta^{(s)})$, $\hat{\mathbf{s}}[n|m]$, and $\mathbf{R}[n|m])$ are given by (A.3), (A.9), and (A.7), respectively. Finally, expressions for $\hat{\mathbf{s}}[n]$ and $\mathbf{R}[n]$ are derived as the mean vector and the covariance matrix of the distribution $\mathrm{p}\Big(\mathbf{s}[n]\Big|\mathbf{x}[1:N], \theta^{(s)}\Big)$, given by (A.10), respectively, i.e.,

$$\begin{aligned}\hat{\mathbf{s}}[n] &= E\left\{E\left\{\mathbf{s}[n]\middle|\phi[n] = m, \mathbf{x}[1:N], \theta^{(s)}\right\}\right\} \\ &= \sum_{m=1}^{M} P_{m,n}(\theta^{(s)})\hat{\mathbf{s}}[n|m], \end{aligned} \tag{A.11}$$

and

$$\begin{aligned}\mathbf{R}[n] &= E\left\{Cov\left\{\mathbf{s}[n]\middle|\phi[n] = m, \mathbf{x}[1:N], \theta^{(s)}\right\}\right\} \\ &\quad + Cov\left\{E\left\{\mathbf{s}[n]\middle|\phi[n] = m, \mathbf{x}[1:N], \theta^{(s)}\right\}\right\} \\ &= \sum_{m=1}^{M} P_{m,n}(\theta^{(s)})\mathbf{R}[n|m] + Cov\{\hat{\mathbf{s}}[n|m]\}, \\ &= \sum_{m=1}^{M} P_{m,n}(\theta^{(s)})(\mathbf{R}[n|m] + \hat{\mathbf{s}}[n|m]\hat{\mathbf{s}}^{\mathrm{T}}[n|m]) - \hat{\mathbf{s}}[n]\hat{\mathbf{s}}^{\mathrm{T}}[n] \end{aligned} \tag{A.12}$$

Appendix B

Proof of Theorem 1 in Chapter 5

It was shown in [115] that if and only if $B(z)$ is irreducible, then there exists a another finite degree matrix $R(z)$ of size $N_T \times N_R$ such that

$$R(z)B(z) = \mathbf{I}_{N_T}.$$

Moreover, it is shown in [114] that if all the conditions in the above theorem are satisfied then there exist a finite-degree matrix $G(z) = R(z)A(z) = \sum_{i=1}^{n_g} \mathbf{G}_i z^{-i}$ such that

$$G(z)H(z) = \mathbf{I}_{N_T},$$

$$\mathbf{G}_0 = (\mathbf{H}_0^{\mathsf{T}}\mathbf{H}_0)^{-1}\mathbf{H}_0^{\mathsf{T}},$$

and

$$n_G \geq n_c + N_T n_b.$$

The main line of the proof is to find a $N_R \times N_R$ matrix $A(z)$ that is a function of $G(z)$ and satisfies (5.39). Further, the absolute term in $A(z)$ is equal to the identity matrix of size $N_R \times N_R$. Assume that $A(z)$ is in the following form

$$A(z) = D_1(z)G(z) + \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}}, \tag{B.1}$$

where the matrices $D_1(z)$ and $\mathbf{D}_2$ are of sizes $N_R \times N_T$ and $N_R \times (N_R - N_T)$, respectively and they are to be determined. Then, the right hand side of (5.39)

will be equal to

$$\left(D_1(z)G(z) + \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}}\right) H(z) = D_1(z) + \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}}H(z).$$

Hence, from (5.39),

$$D_1(z) = \mathbf{H}_0 - \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}}H(z).$$

and

$$A(z) = \left(\mathbf{H}_0 - \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}}H(z)\right) G(z) + \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}}. \tag{B.2}$$

Note that A(z) is of finite degree since both $H(z)$ and $G(z)$ are of finite degree. Moreover, it is not hard to show that

$$P \leq Q + n_g \leq Q + n_c + N_T n_b. \tag{B.3}$$

Therefore, what remains to complete the proof is to find $D_2(z)$ that makes the absolute coefficient in $A(z)$ is the identity matrix. Putting $z = \infty$ in (B.2) and equating to $\mathbf{I}_{N_R}$, we obtain

$$\left(\mathbf{H}_0 - \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}}H_0\right) \mathbf{G}_0 + \mathbf{D}_2\mathbf{D}_2^{\mathsf{T}} = \mathbf{I}_{N_R}. \tag{B.4}$$

Since $\mathbf{G}_0 = (\mathbf{H}_0^{\mathsf{T}}\mathbf{H}_0)^{-1}\mathbf{H}_0^{\mathsf{T}}$, the matrix $\mathbf{H}_0\mathbf{G}_0$ is symmetric, all its eigenvalues are equal to one, and its rank is equal to N_T. Thus, we can express it in the form

$$\mathbf{H}_0\mathbf{G}_0 = \sum_{k=1}^{N_T} \mathbf{u}_k\mathbf{u}_k^{\mathsf{T}}$$

where the vectors $\mathbf{u}_k, k = 1, ..., N_T$ are the eigenvectors of $\mathbf{H}_0\mathbf{G}_0$. Furthermore, since $\mathbf{H}_0\mathbf{G}_0$ is symmetric and positive definite, the eigenvectors can be selected to form an orthonormal basis. Thus, equation (B.4) can be satisfied by setting

$$\mathbf{D}_2 = \begin{bmatrix} \mathbf{u}_{N_T+1} & \mathbf{u}_{N_T+2} & \dots & \mathbf{u}_{N_R} \end{bmatrix}, \tag{B.5}$$

where the vectors $\mathbf{u}_k, k = N_T+1, ..., N_R$ form an orthonormal basis for the subspace orthogonal to the subspace spanned by $\mathbf{u}_k, k = 1, ..., N_T$. Thus, for any transfer matrix $H(z)$, we can use (B.2) and (B.5) to find a finite degree matrix $A(z)$ that satisfies (5.39) and the absolute term in $A(z)$ is the identity matrix. This completes the proof of the theorem.

It is interesting to verify Theorem 1 for example 1. We can set $C(z) = \mathbf{I}_{N_R}$ and $B(z) = H(z)$ since $H(z)$ is irreducible Hence, $n_c = 0$ and $n_b = 1$. According to Theorem 1, $n_g = 2$ and there exists a channel equalizer $A(z)$ with degree $P = 3$. [1] The following matrix $G(z)$ is a finite degree left inverse of H(z).

$$G(z) = \begin{bmatrix} 0.5 & -1 & 0.5 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1.0913 & -0.15 & -0.1913 \\ 0 & 0 & 0 \end{bmatrix} z^{-1} + \begin{bmatrix} 0.8504 & -0.0709 & -0.4252 \\ 0 & 0 & 0 \end{bmatrix} z^{-2}.$$

It is straightforward to show that $\mathbf{D}_2 = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}^{\mathsf{T}} / \sqrt{2}$. Substituting in (B.2), the channel equalizer filter is given by

$$A(z) = \mathbf{I} + \begin{bmatrix} 0.9413 & -0.1 & -0.3413 \\ 0 & 0 & 0 \\ 1.2413 & -0.2 & 0.0413 \end{bmatrix} z^{-1} + \begin{bmatrix} 0.5230 & -0.0259 & -0.3678 \\ 0 & 0 & 0 \\ 1.1778 & -0.1159 & -0.4826 \end{bmatrix} z^{-2} + \begin{bmatrix} -0.2551 & 0.0213 & 0.1276 \\ 0 & 0 & 0 \\ 0.2551 & -0.0213 & -0.1276 \end{bmatrix} z^{-3}.$$

[1] Note that the Theorem 1 guarantees the existence of $A(z)$ if its degree satisfies (5.40). However, for some special problems such as Example 1, we can find some matrices $A(z)$ satisfying (5.39) and their degrees violates the inequality in (5.40).

Appendix C

Convergence Analysis of the EM algorithm used to estimate the equalizer filter

In this appendix, we analyze the convergence behavior of the EM algorithm. The main objective in our analysis is to show that the EM algorithm admits a capture set [81]. That is, if the EM algorithm is initialized with some model parameters $\lambda^{(0)}$ in the domain of attraction of a local (or global) maximizer λ^* of the incomplete log-likelihood function, it is guaranteed that the EM algorithm will converge with a high probability to λ^*. In order to prove this statement, we need to prove that the EM algorithm can be considered as a special case of the quasi-Newton optimization techniques.

Our derivations will be greatly simplified if we combine $\tilde{\mathbf{A}}$ and $\mathbf{H}_0$ into a bigger matrix $\Psi = \begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{H}_0 \end{bmatrix}$ and define

$$\zeta_m[n] \equiv \begin{bmatrix} \mathbf{y}^{\mathsf{T}}[n] & \mathbf{s}_m^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}.$$

It is also convenient to associate with the model parameter string λ a model parameter vector Θ, in the form

$$\Theta = \begin{bmatrix} w_1 & \dots & w_M & \mathsf{vec}\,(\mathbf{\Sigma})^{\mathsf{T}} & \mathsf{vec}\,(\Re(\Psi))^{\mathsf{T}} & \mathsf{vec}\,(\Im(\Psi))^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}},$$

where the vec (.) operator squeezes its matrix argument into one long column vector by concatenating all the columns vertically and in order. In this section, both λ and Θ may be used interchangeably. In addition, we shall express the likelihood function as $\mathcal{L}(\Theta) \equiv \log \mathrm{p}(X|\Theta)$ to simplify the notation.

Given some model parameters $\Theta^{(s)}$, it is required to find a relationship between the new iterate $\Theta^{(s+1)}$ obtained by the EM algorithm and $\frac{\partial \log \mathcal{L}(\Theta)}{\partial \Theta}\Big|_{\Theta=\Theta^{(s)}}$. The main theme of our derivations is to derive expressions for the derivative of the incomplete log-likelihood function with respect to each parameter alone and then employ the EM update equation to find the desired relations. It should be noted that the following constraint was imposed in our derivations of the update equations of the model priors, $w_1, ..., w_M$,

$$\sum_{m=1}^{M} w_m = 1.$$

Hence, for the model priors, we should consider

$$\log \mathcal{L}'(\lambda) = \log \mathcal{L}(\lambda) + \beta \left(\sum_{m=1}^{M} w_m - 1 \right)$$

instead of $\log \mathcal{L}(\lambda)$. The derivative of $\log \mathcal{L}'(\lambda)$ with respect to w_m is given by

$$\frac{\partial \log \mathcal{L}'(\lambda)}{\partial w_m} = \sum_{n=1}^{N} \frac{\mathcal{CN}\left(\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n]; \mathbf{H}_0\mathbf{s}_m; \mathbf{\Sigma}\right)}{Z_n(\lambda)} + \beta,$$

where

$$Z_n(\lambda) = \sum_{m=1}^{M} w_m \mathcal{CN}\left(\mathbf{x}[n] - \tilde{\mathbf{A}}\mathbf{y}[n]; \mathbf{H}_0\mathbf{s}_m; \mathbf{\Sigma}\right).$$

Using (5.50) and performing simple manipulations, we get

$$w_m^{(s+1)} = \frac{w_m^{(s)}}{N} \left(\frac{\partial \log \mathcal{L}'(\lambda)}{\partial w_m}\Big|_{w_m=w_m^{(s)}} - \beta \right).$$

It is not hard to show that $\beta = -N$ in our derivation for the update equation of $w_m^{(s+1)}$. Hence, the above equation simplifies to

$$w_m^{(s+1)} = w_m^{(s)} + \frac{w_m^{(s)}}{N} \frac{\partial \log \mathcal{L}'(\lambda)}{\partial w_m}\Big|_{w_m=w_m^{(s)}}. \tag{C.1}$$

Regarding other parameters, it makes no difference to consider $\log \mathcal{L}(\lambda)$ rather than $\log \mathcal{L}'(\lambda)$. From (5.43), the derivative of $\log \mathcal{L}(\lambda)$ with respect to $\mathbf{\Sigma}$ is given by

$$\frac{\partial \log \mathcal{L}'(\lambda)}{\partial \mathbf{\Sigma}} = \sum_{m,n} P_{n,m}(\lambda) \left(-\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}^{-1} \Re(\mathbf{e}_m[n]\mathbf{e}_m^{\dagger}[n]) \mathbf{\Sigma}^{-1} \right).$$

Substituting (5.49) in the above equation yields the following result

$$\left. \frac{\partial \log \mathcal{L}'(\lambda)}{\partial \mathbf{\Sigma}} \right|_{\mathbf{\Sigma}=\mathbf{\Sigma}^{(s)}} = \left(\sum_{m,n} P_{n,m}(\lambda) \right) \left(-\mathbf{\Sigma}^{(s)-1} + \mathbf{\Sigma}^{(s)-1} \mathbf{\Sigma}^{(s+1)} \mathbf{\Sigma}^{(s)-1} \right)$$

or

$$\mathbf{\Sigma}^{(s+1)} = \mathbf{\Sigma}^{(s)} + \frac{1}{\sum_{m,n} P_{n,m}(\lambda)} \mathbf{\Sigma}^{(s)} \left. \frac{\partial \log \mathcal{L}'(\lambda)}{\partial \mathbf{\Sigma}} \right|_{\mathbf{\Sigma}=\mathbf{\Sigma}^{(s)}} \mathbf{\Sigma}^{(s)}$$

Applying the $\mathsf{vec}\,(.)$ operator to both sides of the above equation, we obtain

$$\mathsf{vec}\left(\mathbf{\Sigma}^{(s+1)}\right) = \mathsf{vec}\left(\mathbf{\Sigma}^{(s)}\right) + \frac{1}{\sum_{m,n} P_{n,m}(\lambda^{(s)}} \left(\mathbf{\Sigma}^{(s)} \otimes \mathbf{\Sigma}^{(s)}\right) \mathsf{vec}\left(\frac{\partial \log \mathcal{L}'(\lambda^{(s)})}{\partial \mathbf{\Sigma}^{(s)}} \right), \tag{C.2}$$

where $\otimes$ denotes the Kronecker product between two matrices. For the derivation of a similar relation for Ψ, it is useful to combine equations (5.47) and (5.48) into the following equation

$$\sum_{m,n} P_{n,m}(\lambda^{(s)}) \left(\mathbf{x}[n] - \Psi^{(s+1)} \zeta_m[n] \right) \zeta_m^{\dagger}[n] = 0. \tag{C.3}$$

In addition, it is useful to utilize Wirtinger definition for complex derivatives in our derivations. It is straightforward to prove that

$$\frac{\partial \log \mathcal{L}'(\lambda^{(s)})}{\partial \Psi^{(s)*}} = \sum_{m,n} P_{n,m}(\lambda^{(s)}) \mathbf{\Sigma}^{(s)-1} \left(\mathbf{x}[n] - \Psi^{(s)} \zeta_m[n] \right) \zeta_m^{\dagger}[n].$$

Again, substituting (C.3) in the above equation and doing simple re-arrangement, we obtain the following relation

$$\Psi^{(s+1)} = \Psi^{(s)} + \mathbf{\Sigma}^{(s)} \frac{\partial \log \mathcal{L}(\lambda^{(s)})}{\partial \Psi^{(s)*}} \left(\sum_{m,n} P_{n,m}(\lambda^{(s)}) \zeta_m[n] \zeta_m^{\dagger}[n] \right)^{-1}$$

or

$$\mathsf{vec}\left(\Psi^{(s+1)}\right) = \mathsf{vec}\left(\Psi^{(s)}\right) + \left(\left(\sum_{m,n} P_{n,m}(\lambda^{(s)}) \zeta_m[n] \zeta_m^{\dagger}[n] \right)^{-1} \otimes \mathbf{\Sigma}^{(s)} \right) \mathsf{vec}\left(\frac{\partial \log \mathcal{L}(\lambda^{(s)})}{\partial \Psi^{(s)*}} \right) \tag{C.4}$$

From (C.1), (C.2), and (C.4), it is obvious that the new model parameter iterate $\Theta^{(s+1)}$ can be expressed in the form

$$\Theta^{(s+1)} = \Theta^{(s)} + \mathbf{W}^{(s)} \left. \frac{\partial \log \mathcal{L}(\Theta)}{\partial \Theta} \right|_{\Theta=\Theta^{(s)}}, \tag{C.5}$$

where

$$\mathbf{W}^{(s)} = \begin{bmatrix} \mathbf{U}_w^{(s)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_{\mathbf{\Sigma}}^{(s)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Re\left(\mathbf{U}_{\Psi}^{(s)}\right) & -\Im\left(\mathbf{U}_{\Psi}^{(s)}\right) \\ \mathbf{0} & \mathbf{0} & \Im\left(\mathbf{U}_{\Psi}^{(s)}\right) & \Re\left(\mathbf{U}_{\Psi}^{(s)}\right) \end{bmatrix}$$

,

$$\mathbf{U}_w^{(s)} = \frac{1}{N}\mathrm{diag}(w_1^{(s)}, \ldots, \mathbf{w}_M^{(s)}),$$

$$\mathbf{U}_{\mathbf{\Sigma}}^{(s)} = \frac{1}{\sum_{m,n} P_{n,m}(\lambda^{(s)}} \left(\mathbf{\Sigma}^{(s)} \otimes \mathbf{\Sigma}^{(s)}\right),$$

$$\mathbf{U}_{\Psi}^{(s)} = \left(\left(\sum_{m,n} P_{n,m}(\lambda^{(s)}) \zeta_m[n] \zeta_m^{\dagger}[n] \right)^{-1} \otimes \mathbf{\Sigma}^{(s)} \right)$$

Note that the matrix $\mathbf{W}^{(s)}$ is positive definite. Hence, the EM algorithm belongs to the quasi-Newton optimization methods. According to [81], if Θ^* is the only stationary point of the incomplete log-likelihood function in the some open set and if there exists a constant C such that the maximum singular value of $\mathbf{W}^{(s)}$ is less than C for all s, then there exists an open set S containing Θ^* such that if $\Theta^{(s_0)} \in S$ for some s_0 then $\Theta^{(s)} \in S$ for all $s \geq s_0$. Moreover, the sequence $\{\Theta^{(s)}\}$ converges uniformly to Θ^*. That is, the EM update equations admit a capture set for the incomplete log-likelihood function. In addition, equation (C.5) can be used to monitor the convergence of the Em algorithm.

Bibliography

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans.Automatic Control*, 19:716–723, 1974. 51

[2] S. An and Y. Hua. Blind signal separation and blind system identification of irreducible mimo channels. *Signal Processing and its Applications, Sixth International, Symposium on.2001*, 1:276–279, 2001. 70

[3] J. M. M. Anderson. A generalized likelihood ratio test for detecting land mines using multispectral images. *Geoscience and Remote Sensing Letters, IEEE*, 5(3):547–551, July 2008. 67

[4] P. Angkititrakul and J. H. L. Hansen. Discriminative in-set/out-of-set speaker recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(2):498–508, Feb. 2007. 94

[5] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, 1 2000. 4, 27, 28

[6] M. M. H. El Ayadi, M. S. Kamel, and F. Karray. Speech emotion recognition using gaussian mixture vector autoregressive models. *ICASSP 2007*, 4:957–960, 2007. 5, 87

[7] Moataz M. H. El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Modeling and equalizing multiple-input-multiple-output channels using vector autoregressive gaussian mixture models. *submitted to Signal Processing*. 60

[8] L. Bahl, P. Brown, P. de Souza, and R. Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition, 1986. ID: 1. 17, 109

[9] C. Barras and J. L Gauvain. Feature and score normalization for speaker verification of cellular data, 2003. 4, 27

[10] P. Bianchi and P. Loubaton. Identification and deconvlution of multichannel linear non-gaussian. *IEEE Trans.Signal Processing*, 45(3):268–271, 1997. 70

[11] P. Bianchi and P. Loubaton. On the blind equalization of continuous phase modulated signals. *IEEE Trans.Signal Processing*, 55(3):1047–1061, 2007. 6, 70

[12] S. Bidon, O. Besson, and J.-Y. Tourneret. The adaptive coherence estimator is the generalized likelihood ratio test for a class of heterogeneous environments. *Signal Processing Letters, IEEE*, 15:281–284, 2008. 5, 67

[13] C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995. 21, 23, 35, 51, 90

[14] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans.Acoust.Speech Signal Process.*, 27(2):113–120, 1979,. 4, 25

[15] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss. A database of german emotional speech. *Proc.Interspeech 2005, Lissabon, Portugal*, 2005. 5, 88

[16] J. A. Cadzow. Blind deconvolution via cumulant extrema. *IEEE Signal Processing Mag.*, 13:24–42, 1996. 70

[17] J. P. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997. 2, 4, 19

[18] J.P. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, Sep 1997. 2

[19] J. E. Cavanaugh. A large-sample model selection criterion based on kullbacks symmetric divergence. *Stat.Prob.Lett.*, 42:333–343, 1999. 52

[20] B. Chen and A. P. Petropulu. Frequency domain blind mimo system. *IEEE Trans.Signal Processing*, 49(8):1677–1688, 2001. 70

[21] C. Y. Chi and C. H. Chen. Cumulant-based inverse filter criteria for mimo blind deconvolution:. *IEEE Trans.Signal Processing*, 49(7):1282–1299, 2001. 70

[22] C. Y. Chi, C. Y. Chen, C. H. Chen, C. C. Feng, and C. H. Peng. Blind identification of simo systems and simultaneous estimation of multiple time delays. *IEEE Trans.Signal Processing*, 52(10):2749, 2004. 70

[23] C. Y. Chi, C. C. Feng, C. H. Chen, and C. Y. Chen. *Blind Equalization and System Identification. Batch Processing: Algorithms, Performance, and Applications*. Springer, 2006. 70

[24] H. A. Cirpan and M. Tsatsanis. Blind receivers for nonlinearly modulated. *IEEE Trans.Signal Processing*, 47(10):583–586, 1999. 6, 71

[25] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 28(4):357–366, Aug 1980. 14

[26] J. de Veth and L. Boves. Phase-corrected rasta for automatic speech recognition over the phone. *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, 2:1239–1242 vol.2, Apr 1997. 26

[27] M. DeGroot. *Optimal statistical decisions*. New York:McGraw-Hill, 1970. 39

[28] John R. Deller, John G. Proakis, and John H. Hansen. *Discrete-Time Processing of Speech Signals*. McMillan, 1993. 14

[29] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J.Royal Stat.Soc.*, 39:1–38, 1977. 6, 20, 32, 71

[30] J. Diggle and I. Wasel. Spectral analysis of replicated biomedical time series. *Appl.Statist.*, 46:31–71, 1997. 42

[31] Jasha Droppo, Alex Acero, , and Li Deng. A nonlinear observation model for removing noise from corrupted speech log mel-spectral energies. *ICSLP02*, pages 1569–1572, 2002. 25

[32] A. Drygajlo and M. El-Maliki. Speaker verification in noisy environments with combined spectral subtraction and missing feature theory. *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, 1:121–124, May 1998. 25

[33] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000. 17, 23, 87

[34] Y. Ephraim and N. Merhav. Hidden markov processes. *Information Theory, IEEE Transactions on*, 48(6):1518–1569, Jun 2002. 20

[35] Y. Ephraim and W. Roberts. Revisiting autoregressive hidden markov modeling of speech signals. *IEEE Signal Processing Letters*, 12(2):166–169, 2005. 36, 43

[36] K. R. Farrell, R. J. Mammone, and K. T. Assaleh. Speaker recognition using neural networks and conventional classifiers, 1994. 2, 21

[37] K.R. Farrell. Text-dependent speaker verification using data fusion. *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, 1:349–352 vol.1, May 1995. 21

[38] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice including a complexity analysis. *Mathematics of Computation*, 44(170):463–471, April,1985. 111

[39] S. Fine, J. Navratil, and R.A. Gopinath. A hybrid gmm/svm approach to speaker identification. *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, 1:417–420 vol.1, 2001. 23

[40] E. Fisher, J. Tabrikian, and S. Dubnov. Generalized likelihood ratio test for voiced-unvoiced decision in noisy speech using the harmonic model. *IEEE Trans.Audio, Speech & Language Processing*, 14(2):502–510, March 2006,. 5, 68

[41] S. Furui. Cepstral analysis technique for automatic speaker verification. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 29(2):254–272, Apr 1981. 25

[42] S. Furui. Recent advances in speaker recognition. *Pattern Recognition Letters*, 18:859–872(14), September 1997. 2

[43] M. J. F. Gales and S. J. Young. Cepstral parameter compensation for hmm recognition in noise. *Speech Communication*, 12:231–240, 1993,. 4, 26

[44] M. J. F. Gales and S. J. Young. Cepstral parameter compensation for hmm recognition in noise. *Speech Commun.*, 12(3):231–239, 1993. 27

[45] S. Gannot, D. Burshtein, and E. Weinstein. Iterative and sequential kalman filter-based speech enhancement algorithms. *Speech and Audio Processing, IEEE Transactions on*, 6(4):373–385, Jul 1998. 114

[46] F. Gao and A. Nallanathan. Resolving multidimensional ambiguity in blind channel estimation of mimo-fir systems via block precoding. *IEEE Trans.Vehicular Technology*, 57(1):11–21, 2008. 70, 73, 79

[47] J. L Gauvain and C. H Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains, 1994. ID: 1. 39

[48] H. Gish, M. Krasner, W. Russell, and J. Wolf. Methods and experiments for text-independent speaker recognition over telephone channels. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, 11:865–868, Apr 1986. 3

[49] J. Gudnason and M. Brookes. Voice source cepstrum coefficients for speaker identification. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4821–4824, 31 2008-April 4 2008. 94

[50] V. Guruswami. List decoding from erasures: bounds and code constructions. *Information Theory, IEEE Transactions on*, 49(11):2826–2833, Nov. 2003. 11

[51] J. D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, 1994. 44, 45

[52] B. Hassibi and B. M. Hochwald. How much training is needed in multiple-antenna wireless links? *IEEE Trans.Information Theory*, 49(4):951–963, 2003. 70

[53] Jialong He, Li Liu, and G.; Palm. A discriminative training algorithm for vq-based speaker identification. *Speech and Audio Processing, IEEE Transactions on*, 7(3):353 – 356, May 1999. 19

[54] Larry P. Heck, Yochai Konig, M. Kemal Sönmez, and Mitch Weintraub. Robustness to telephone handset distortion in speaker recognition by discriminative feature design. *Speech Commun.*, 31(2-3):181–192, 2000. 25

[55] P. Henrici. *Applied and Computational Complex Analysis*, volume III. New York:Wiley, 1986. 76

[56] H. Hermansky and N. Morgan. Rasta processing of speech. *Speech and Audio Processing, IEEE Transactions on*, 2(4):578–589, Oct 1994. 26

[57] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. Compensation for the effect of the communication channel in auditory-like analysis of speech (rasta-plp). *Proceedings European Conf. on Speech Communication and Technology. EUROSPEECH*, pages 1367–1370, 1991. 25

[58] A.L. Higgins, L.G. Bahler, and J.E. Porter. Voice identification using nearest-neighbor distance measure. *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, 2:375–378 vol.2, Apr 1993. 18

[59] M. Hsken and S. Peter. Recurrent neural networks for time series classification. *Neurocomputing*, 50:223–235, 2003. 88, 91

[60] M. Hunt. Further experiments in text-independent speaker recognition over communications channels. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '83.*, 8:563–566, Apr 1983. 3

[61] Y. Inouye and K. Tanebe. Super exponential algorithms for multichannel blind deconvolution. *IEEE Trans.Signal Processing*, 48(3):881–888, 2000. 70

[62] Qin Jin. *Robust Speaker Recognition.* PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, January 2007. 3

[63] N. L. Johnson and S. kotz. *Distributions in Statistics.* New York:Wiley, 1972. 38

[64] Biing-Hwang Juang, Wu Hou, and Chin-Hui Lee. Minimum classification error rate methods for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 5(3):257–265, May 1997. 17, 109

[65] G. K. Kaleh and R.Vallet. Joint parameter estimation and symbol detection. *IEEE Trans.Communication*, 42(10):2406–2413, 1994. 6, 71

[66] S. Kay and D. Sengupta. Optimal detection in colored non-gaussian noise with unknown parameters. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, 12:1087–1090, Apr 1987. 68

[67] L. G. Kersta. Voiceprint identification. *Nature*, 196:12531257, 1962. 2

[68] K. Kohno, Y. Inouye, and M. Kawamoto. Robust super-exponential methods for blind equalization of mimo-iir systems. *ICASSP 2006*, 5, 2006. 70

[69] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464 – 1480, Apr 1990. 19

[70] Ulrich H.-G. Kressel. Pairwise classification and support vector machines. *Advances in kernel methods: support vector learning*, pages 255–268, 1999. 23

[71] K. Lee and J. Lee. Recognition of noisy speech by a nonstationary ar hmm with gain adaptation under unknown noise. *IEEE Trans.Speech & Audio Processing*, 9(7):741–746, 2001. 43

[72] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design, 1980. ID: 1. 19, 35

[73] R.P. Lippmannl. Review of neural networks for speech recognition. *Neural Computation*, 1:1–38, 1989. 21

[74] D. Luenberger. *Linear and Nonlinear Programming, 2nd ed. Reading.* MA: Addison Wesley, 1984. 48

[75] M.W. Mak, W.G. Allen, and G.C. Sexton. Speaker identification using radial basis functions. *Artificial Neural Networks, 1993., Third International Conference on*, pages 138–142, May 1993. 21

[76] S. L. Marple. *Digital Spectral Analysis with Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1987. 77

[77] M. Martone. Non-gaussian multivariate adaptive ar estimation using the super exponential algorithm. *IEEE Trans.Signal Processing*, 44(10):2640–2644, 1996. 70

[78] J. M. Mendel. Tutorial on higher order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305, 1991. 70

[79] J. Ming, T. J. Hazen, J. R. Glass, and D. A. Reynolds. Robust speaker recognition in noisy conditions, 2007. 2, 4

[80] D.P. Morgan and C.L. Sco eld. *Neural Networks and Speech Processing*. Kluwer Academic Publishers, Dordrecht, 1991. 21

[81] H. Nguyen and B. C. Levy. Blind and semi-blind equalization of cpm signals with the emv algorithm. *IEEE Trans.Signal Processing*, 51(10):2650–2664, 2003. 6, 71, 76, 120, 123

[82] H. Nguyen and B. C. Levy. The expectation-maximization viterbi algorithm for blind adaptive channel equalization. *Communications, IEEE Transactions on*, 53(10):1671–1678, Oct. 2005. 71

[83] Geng-Xin Ning, Shu-Hung Leung, Kam-Keung Chu, and Gang Welt. A parallel model combination scheme with improved delta parameter compensation. *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 4 pp.–, May 2006. 27

[84] J. Oglesby and J.S. Mason. Optimisation of neural models for speaker identification. *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, 1:261–264, Apr 1990. 22

[85] A. Panda, N. Tripathi, and T. Srikanthan. Improved spectral subtraction technique for text-independent speaker verification. *Digital Signal Processing, 2007 15th International Conference on*, pages 595–598, July 2007. 25

[86] J. Pelecanos and S. Sridharan. Short-time gaussianization for robust speaker verification. *Proc. ISCA Workshop Speaker Recognition,*, page 213218, 2001. 25

[87] B.L. Pellom, R. Sarikaya, and J.H.L. Hansen. Fast likelihood computation techniques in nearest-neighbor based search for continuous speech recognition. *Signal Processing Letters, IEEE*, 8(8):221–224, Aug 2001. 110

[88] J. Picone. Continuous speech recognition using hidden markov models. *ASSP Magazine, IEEE*, 7(3):26–41, Jul 1990. 19

[89] A. Poritz. Linear predictive hidden markov models and the speech signals. *ICASSP 1982*, pages 1291–1294, 1982. 20, 43

[90] T.F. Quatieri, D.A. Reynolds, and G.C. O'Leary. Magnitude-only estimation of handset nonlinearity with application to speaker recognition. *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, 2:745–748 vol.2, May 1998. 27

[91] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986. 19, 20, 88, 92

[92] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, New Jersey, 1993. 9, 12

[93] L. Rabiner and R. Schafer. *Digital Processing of Speech Signals*. Pearson Education, 1978. 11

[94] Ravi P. Ramachandran, Kevin R. Farrell, Roopashri Ramachandran, and Richard J. Mammone. Speaker recognition general classifier approaches and data fusion methods. *Pattern Recognition*, 35(12):2801–2821, 12 2002. 2, 16, 21

[95] G. Reinsel. *Elements of Multivariate Time Series Analysis*. Springler-Verlag, 1993. 43

[96] D. Reynolds and C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans.Speech and Audio Processing*, 3(1):72–83, 1995. 3, 4, 9, 11, 20, 23, 26, 34, 43, 94

[97] D.A. Reynolds. Large population speaker identification using clean and telephone speech. *Signal Processing Letters, IEEE*, 2(3):46–48, Mar 1995. 94

[98] D.A. Reynolds. An overview of automatic speaker recognition technology. *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, 4:IV–4072–IV–4075 vol.4, 2002. 2

[99] Douglas A. Reynolds, George R. Doddington, Mark A. Przybocki, and Alvin F. Martin. The nist speaker recognition evaluation - overview methodology, systems, results, perspective. *Speech Commun.*, 31(2-3):225–254, 2000. 82, 83

[100] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, January 2000. 4, 24, 26, 27, 34, 85

[101] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–571, 1978. 52

[102] W.J.J. Roberts, Y. Ephraim, and H.W. Sabrin. Speaker classification using composite hypothesis testing and list decoding. *Speech and Audio Processing, IEEE Transactions on*, 13(2):211–219, March 2005. 11

[103] R. C. Rose, E. M. Hofstetter, and D. A. Reynolds. Integrated models of signal and background with application to speaker identification in noise. *IEEE Trans.Speech & Audio Processing*, 2(2):245–257, April 1994,. 3, 4, 25, 30, 34, 37, 53

[104] A. Scherb, V. Kuhn, and K. Kammeyer. Blind identification and equalization of ldpc-encoded mimo systems. *Proc.IEEE 61st Vehicular Technology Conference (VTC)*, 1:562–566, 2005. 98, 101

[105] B. Schuller, G. Rigoll, and M. Lang. Speech emotion recognition combining acoustic features. *Proc.ICASSP 2004*, 1:577–580, 2004. 87, 88

[106] A. K. Seghouane. Vector autoregressive model-order selection from finite samples using kullback's symmetric divergence. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(10):2327–2335, Oct. 2006. 42

[107] J.C. Segura, C. Benitez, A. de la Torre, A.J. Rubio, and J. Ramirez. Cepstral domain segmental nonlinear feature transformations for robust speech recognition. *Signal Processing Letters, IEEE*, 11(5):517–520, May 2004. 25

[108] A. Sharma, S.P. Singh, and V. Kumar. Text-independent speaker identification using backpropagation mlp network classifier for a closed set of speakers. *Signal Processing and Information Technology, 2005. Proceedings of the Fifth IEEE International Symposium on*, pages 665–669, Dec. 2005. 21

[109] I. E. Telatar. Capacity of multi-antenna gaussian channels. *European Trans.Telecommunications*, 10(6):585–595, 1999. 70

[110] R. Teunen, B. Shahshahani, and L. Heck. A model-based transformational approach to robust speaker recogntion. *ICSLP 2000*, 2:495–498, 2000. 4, 26

[111] N.Z Tisby. On the application of mixture ar hidden markov models to text independent speaker recognition. *Signal Processing, IEEE Transactions on*, 39(3):563–570, Mar 1991. 20

[112] L. Tong and S. Perreau. Multichannel blind identification: From subspace to maximum likelihood methods. *Proceedings of the IEEE*, 86(10):1951–1968, 1998. 70

[113] L. Tong, G. Xu, and T. Kailath. Blind identification and equalization of multipath channels. Record of the 25th Asilomar Conference on Signals, Systems, and Computers, Pacific, 1991. 70

[114] J. Tugnait and B. Huang. On a whitening approach to partial channel estimation and blind equalization of fir/iir. *IEEE Trans.Signal Processing*, 48(3):832–845, 2000. 6, 42, 72, 79, 98, 99, 117

[115] J. K. Tugnait. Fir inverses to mimo rational transfer functions with application to blind equalization. *Signals, Systems and Computers, 1996.1996 Conference Record of the Thirtieth Asilomar Conference on*, 1:295–299, 3-6 Nov 1996. 117

[116] J. K. Tungait. Blind spatio-temporal equalization and impulse response. *IEEE Trans.Signal Processing*, 45(1):268–271, 1997. 6, 70

[117] J. K. Tungait, L. Tong, and Z. Ding. Single user channel estimation and equalization. *IEEE Signal Processing Magazine*, 17(3):17–28, 2000. 6, 70

[118] J. Va, I. Santamara, and J. Prez. Deterministic cca-based algorithms for blind. *IEEE Trans.Signal Processing*, 55(7):3867–3878, 2007. 70

[119] V. Vapnik. *Statistical Learning Theaory*. New York:Wiley, 1998. 23

[120] S. Verbout, J. Ooi, J. Ludwig, and A. Oppenheim. Parameter estimation for autoregressive gaussian-mixture model: the emax algorithm. *IEEE Trans.Signal Processing*, 46(10):2744–2756, 1998. 6, 43, 45, 48, 71

[121] Jia-Ching Wang, Chung-Hsien Yang, Jhing-Fa Wang, and Hsiao-Ping Lee. Robust speaker identification and verification. *Computational Intelligence Magazine, IEEE*, 2(2):52–59, May 2007. 21, 23, 24

[122] X. G. Xia, W. Su, and H. Liu. Filterbank precoders for blind equalization:. *IEEE Trans.Circuits Syst.I*, 46(2):19–29, 2001. 73

[123] Bing Xiang, U.V. Chaudhari, J. Navratil, G.N. Ramaswamy, and R.A. Gopinath. Short-time gaussianization for robust speaker verification. *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, 1:I–681–I–684 vol.1, 2002. 25

[124] K. L. Yeung and S. F. Yau. A cumulant-based super-exponential algorithm for blind deconvolution of multi-input. *Signal Processing*, 67(2):141–162, 1998. 70

[125] N.B. Yoma and M. Villar. Speaker verification in noise using a stochastic version of the weighted viterbi algorithm. *Speech and Audio Processing, IEEE Transactions on*, 10(3):158–166, Mar 2002. 25

[126] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK book (for version 3.1)*. 2002. 89

[127] Xicai Yue, Datian Ye, Changxun Zheng, and Xiaoyu Wu. Neural networks for improved text-independent speaker identification. *Engineering in Medicine and Biology Magazine, IEEE*, 21(2):53–58, Mar/Apr 2002. 21

[128] I. Zeljkovic, P. Haffner, B. Amento, and J. Wilpon. Gmm/svm n-best speaker identification under mismatch channel conditions. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4129–4132, 31 2008-April 4 2008. 21, 23

[129] Xu Zhao and Mike Davies. A feasible blind equalization scheme in large constellation mimo systems. *Acoustics, Speech and Signal Processing, 2008.ICASSP 2008.IEEE International Conference on*, pages 1845–1848, March 31 2008-April 4 2008. 111

[130] G. Zhou and W.B. Mikhael. Speaker identification based on adaptive discriminative vector quantisation. *Vision, Image and Signal Processing, IEE Proceedings* -, 153(6):754–760, Dec. 2006. 4, 19