The Role of Dynamic Features in Speaker Verification

A THESIS

Submitted for the Degree of

Doctor of Philosophy

by

Ying Liu

Electrical Engineering

University of Birmingham

November 2009

Acknowledgments

First and foremost I offer my sincerest gratitude to my supervisor, Professor Martin Russell, who has supported me throughout my project with his knowledge and patience. He showed me different ways to approach a research problem. He also showed me the need to be persistent to accomplish any goal in research. Martin was greatly helpful during my writing up of this thesis as well. He proofread and marked up my chapters, and asked me good questions to help me think through my problems. He had confidence in me when I doubted myself. I attribute the level of my PhD degree to his encouragement and constant guidance and without him this thesis, too, would not have been completed.

I am heartily thankful to Professor Michael Carey, who has offered advice and suggestions throughout my work on this project.

My sincere thanks are due to the official examiners, Dr. Thomas Hain and Dr. Peter Jancovic, for their detailed review, constructive criticism and advice during the preparation of this thesis.

I also wish to thank Eric Hansen and Tim Anderson from the Air Force Research Laboratory (AFRL), for their collaboration on my speaker verification experiments on Switchboard and for providing their conventional background GMM used for analysis of the delta parameters in this thesis.

This work was partly funded by EOARD (European Office of Aerospace Research and Development) award FA8655-03-1-3060 and by EPSRC grant EP/C515986/1 "A Unified Model for Speech Recognition and Synthesis".

I have been aided for many years in running the equipment by Dr Sridhar Pammu, a fine technician who helped me with all sorts of problems on the SUN station, the computer 'cluster' in our laboratory on which I ran various experiments, and my personal computer.

My warm thanks to Mrs Mary Winkles from the Postgratuate Office, for her help and support throughout my PhD study.

In my daily work I have been blessed with a friendly and cheerful group of fellow students. Simon Smith essentially taught me everything I know about unix when I started my PhD. Michael Wong helped me set up my first speaker verification system and patiently answered lots of questions from me. Philip Jackson and Bo-Hoi Lo both helped me understand my subject with their knowledge and PhD experiences. Shona D'arcy has fascinated me with her energy and hard work. Shona also organized all the group activities for us. Xiaoyan Zheng and Hongwei Hu had been doing projects closely related to mine so we talked about them a lot. We also became close friends. I also want to thank the people in room 422, Paul, Xin, Neil, James, Abualseoud and Gheida. You have been a pleasure to work with.

Finally I want to thank my family. I am forever indebted to my parents,

Siqin and Furong, for giving me life in the first place, for sparing no effort to provide the best possible environment for me to grow up, for unconditional support and encouragement to pursue my interests, even when the interests went beyond boundaries of language and geography. I am grateful to my husband Yi for always being there for me. Special thanks to my little boys Han and Tian. Your love has been a powerful source of inspiration and energy. Also thanks to my parents-in-law, Yuan'e and Xiying, for loving me as your own daugher and always believing in me. This thesis is simply impossible without the love and support from you all.

Abstract

The thesis presents experiments and analyses to explore the role of dynamic features in speaker verification (SV). The project is based on the theory that dynamic information in speech should contain important speaker information, thus modelling the dynamic information into speaker verification systems should have the potential to improve the SV performance.

Firstly trajectory-based segmental hidden Markov models (SHMMs) were used to explore the utility of modelling speech dynamics in Text-Independent (TI) and Text-Dependent Speaker Verification (TD-SV). Experiments on TD-SV using SHMMs on the YOHO database show performance improvement. However there is no significant improvement for TI-SV from experiments on the Switchboard database, using segmental HMMs. Analysis of the TD-SV results confirms that the speech dynamics modeled by segmental HMMs contribute more to the speaker verification accuracy. Analysis of the TI-SV results indicates that the lack of speech dynamic information is a feature of both the segmental GMM system and a conventional GMM

system. It seems that the priority of the maximum likelihood (ML) training algorithm is to model stationary regions, and the role of dynamic features, or of the differential parameters in conventional GMM system, is to ensure that the classification focuses on static regions rather than to model dynamic regions.

Next experiments and analyses on TI-SV were carried out using conventional GMMs. We compare the models and verification performance obtained with 'delta' MFCC parameters alone with the more conventional 'static-plus-delta' parameters. Without RASTA filtering, the 'delta-only' system works best. However, after RASTA filtering, the performance of the 'static-plus-delta' system performs best. The results suggest that the good performance of the 'delta-only' system before RASTA is mainly due to the noise robustness of the delta parameters. Unfortunately, the scores obtained with the 'delta only' and 'static plus deltas' systems are highly correlated, and a fused system gives little improvement over the 'statics plus deltas' system.

Novel Contributions

This Thesis includes the following novel contributions:

- Successfully applied segmental HMMs to TD-SV on YOHO.
- Demonstrated that modelling dynamics in TD-SV contributes to the SV accuracy.
- Applied segmental GMMs to TI-SV on Switchboard.
- Successfully reduced computational load for recognition using the segmental GMM system.
- Developed and evaluated state-of-the-art GMM based TI-SV systems on Switchboard with different parameter sets: statics only, deltas only, and statics plus deltas in order to better understand the role of dynamics.

List of Notations

p probability density function

$\mathcal{L}$ the log likelihood

$Y = (y_1, y_2, ..., y_t, ..., y_T)$ a set of feature observations, where y_t is a D-dimensional observed feature vector.

$X = (x_1, x_2, ..., x_t, ..., x_T)$ a state sequence in an HMM, where x_t $\sigma_1, \sigma_2, ..., \sigma_N$, $\sigma_i (i = 1, 2, ..., N)$ is one of N states in the model.

W a particular word produced by a speaker

UBM the universal background model

S speaker verification score

M the number of Gaussian mixture components in a GMM or segmental SGMM

μ_i the mean of the ith state (in HMM) or Gaussian component (in GMM)

Σ_i the variance of the ith state or Gaussian component

ω_i the weight of the ith Gaussian mixture component

$b_m(y)$ the pdf of the mth state or Gaussian component

a_{ij} the transition probability from state i to state j

$\alpha_i(t)$ the forward probability in Baum-Welch algorithm

$\beta_j(t)$ the backward probability in Baum-Welch algorithm

$\mathbf{f}$ the trajectory in a segmental HMM or GMM

$\mathbf{m}$ the slope of the trajectory f of the segmental model

$\mathbf{c}$ the mid point of the trajectory f of the segmental model

τ_{max} the maximum duration of the segmental model

Δ the first derivatives of the speech features

Δ^2 the second derivatives of the speech features

λ_1 the language model scale factor

λ_2 the token insertion penalty

Contents

List of Figures

CHAPTER 1

Introduction

1.1 The Problem Formulation

Speaker recognition is the process of recognizing a speaker on the basis of individual information included in his or her speech signal. This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, telephone banking, and security control for confidential information areas. There are also important applications in forensics (Meuwly and Drygajlo 2001). With characteristics such as biometric particularity and convenience of use, speaker recognition naturally becomes the favorable approach to solving the problems of unauthorized entry to services with control access.

Speech is the signal produced initially from a brain message, which is then transmitted through several different stages or levels: linguistic, physiological or articulatory, and acoustic. The huge variation of these speaker-related speech signals is a result of a combination of the inherent physiological differ-

ences in the vocal tract, the different individual learned speaking habits, and the intra-speaker differences due to many factors such as sickness or different emotional states. In speaker recognition, features or methods which exhibit low intra-speaker variation and high inter-speaker variation are desired.

The object of the research described in this thesis was to use Segmental Hidden Markov models (SHMMs) as a vehicle to analyze the significance of dynamics for speaker recognition. We believe that the dynamic information in speech contains speaker information and individual differences which should help increase performance of speaker verification systems. Some evidences are shown in Chapter 2.4. By applying segmental HMMs into speaker verification, we try to update the current speaker verification systems with modelled dynamic information. By investigating and analyzing the experiments' results we also try to understand the role of speech dynamics in text-dependent and -independent speaker verification.

1.2 Scope of Thesis

The thesis is organized as follows. In Chapter 2 some background knowledge about speaker recognition and methods are described, including the pattern matching speaker recognition methods and the popularly used speaker recognition corpora. The relationship of speech dynamics and speaker verification, as well as the motivation for the Segmental HMMs are also included. Chapter 3 presents front-end processing for speaker recognition. Chapter 4 introduces the stochastic modelling methods which are the state-of-the-art approaches to speaker verification. The Gaussian Mixture Model (GMM) and the Hid-

den Markov Model (HMM) are described as well as the maximum likelihood procedure for the stochastic model parameter estimation. Other auxiliary methods such as speaker normalization and speaker adaptation techniques are also introduced. Chapter 5 presents the theory of segmental HMMs and introduces the Linear Trajectory Segmental HMMs, the specific model which is used in this thesis.

In Chapter 6 the Text-Dependent Speaker Verification (TD-SV) experiments using SHMMs on the YOHO database are described. The results show that the SHMM system outperforms the conventional HMM system by using the linear trajectory segmental models. The segmental version of a GMM Text-Independent Speaker Verification (TI-SV) system built on Switchboard is presented in Chapter 7. The results do not show any benefit from using the segmental models. An analysis of the Switchboard experiments results is given in Chapter 8, followed by an analysis of the YOHO results in Chapter 9. Chapter 10 describes a set of experiments using conventional GMM systems each of which has a different parameter set, to help us investigate the role of delta parameters in TI-SV. Experiments and results of fusing the systems described in Chapter 10 are presented in Chapter 11. Finally, conclusions and discussions are presented in Chapter 12.

CHAPTER 2

Background

Speaker recognition is a technique which recognizes speaker identity. Although the target of speaker recognition tasks is different from speech recognition, which aims to recognize the contents of a speech signal, many of their techniques are similar. In this chapter some background knowledge of speaker recognition will be introduced, including the common methods and speech corpora often chosen for speaker recognition experiments. The key issue of this project, the potential benefit of employing speech dynamics in speaker recognition will be discussed in the final part of this chapter.

2.1 Speaker Recognition

Speaker recognition tasks can be divided into two types: speaker identification (SI) and speaker verification (SV). Speaker identification is the process of determining which of a set of registered speakers spoke a given utterance. Speaker verification is the process of accepting or rejecting the identity claim

of a speaker based on the given utterance. This project is focused on speaker verification systems. Depending on whether the contents of the input utterances are known in advance, speaker recognition can be further divided into text-dependent and text-independent methods. In a text-dependent process the text to be spoken by the user is 'known' by the system, while in a text-independent process there are no constraints on the text so the system must be able to process utterances of any text.

The general approach to speaker recognition consists of several main steps: feature extraction, pattern matching, speaker reference models generating, and decision making. The first step, which will be described in detail in Chapter 3, is to extract from the input utterance a sequence of feature vectors which contains and idealy only contains information necessary for the recognition task. A comparison of these feature vectors is then made with the reference templates in which case a distance is evaluated, or statistical models in which case a probability is evaluated. The last step is to make a recognition decision based on the distances or probabilities.

In the following sections the basic pattern-matching approaches and the commonly used speech corpora for speaker recognition will be introduced. I will also discuss some evidence that the dynamic information in the speech could contain individual differences. Thus if the dynamic information is embedded into the speaker recognition system, it was hoped to improve speaker recognizer performance. This formed the initial motivation of my project.

2.2 Pattern-matching Methods

Two pattern-matching approaches to text-dependent speaker recognition are commonly used. One is Dynamic Time Warping (DTW) spectral template matching, an implementation proposed by Furui (Furui 1981). In this approach, each utterance (usually word-level utterance) is represented by a sequence of feature vectors, which is time-aligned with a reference feature vector sequence. The distance between the utterance and the reference is then calculated.

An alternative approach is Hidden Markov Model (HMM) based statistical modelling (Bahl et al. 1983; Rabiner 1989), in which each utterance (usually phone-like sub word unit) is represented as an HMM. An estimate of the probability of the utterance given the model is calculated using the Viterbi algorithm (Viterbi 1967). Based on Bayes' rule, the posteriori probability of the model given the utterance can then be calculated. The HMM has the capability of efficiently modelling statistical variation in spectral features and hence has been commonly used in text-dependent speaker recognition. In general, HMM-based methods have achieved better recognition accuracies than the DTW-based methods (Zheng and Yuan 1988; Rosenberg et al. 1991).

Approaches to text-independent speaker recognition include vector quantization (VQ) (Soong et al. 1985), neural networks (Farrell et al. 1994) and Gaussian mixture model (GMM) -based statistical modelling (Rose and Reynolds 1990; Reynolds 1992; Reynolds and Rose 1995). Of the above, the dominant approach over recent years is the GMM. Each GMM emitting state

is connected to a non-emitting initial and final states with a transition from the initial state with probability represented by a weight and a transition to the final state with probability 1. Each component is associated with a Gaussian output distribution and a weight.

In this project the main methods are the stochastic methods. The HMMs are employed for text-dependent, and GMMs are applied to text-independent speaker verification. The details of the stochastic methods including HMMs and GMMs are described in Chapter 4.

2.3 Speaker Recognition Corpora

The use of standard speech corpora for development and evaluation of speaker recognition systems is very important, because they enable formal assessment and comparison of systems, and hence measurement of progress. Commonly used standard corpora include the TIMIT speech recognition corpus (Garofolo et al. 1993), the King corpus (Consortium 1992), the YOHO speaker verification corpus (Higgins 1990), the Switchboard corpora (Campbell and Reynolds 1999) and the OGI speaker recognition corpus (Cole et al. 1998). In my project I used the TIMIT, YOHO and Switchboard corpora.

The TIMIT corpus is very well-known, and comprises recordings of read speech. TIMIT was designed for developing and evaluating automatic speech recognition systems. As it has a large number of speakers, it has been used for speaker recognition studies. TIMIT is labeled at the phone level, and is therefore particularly useful for building phone-level acoustic models.

The YOHO Speaker-Verification corpus was collected by ITT under a US

government contract and was designed to develop and test speaker verification systems in office environments with limited vocabulary. The YOHO database was the first large-scale, scientifically controlled and collected, high-quality speech database for speaker-verification testing at high confidence levels. There are only low level office and computer noises in the data. It was chosen in my study because of its established use in text-dependent speaker verification. The corpus comprises recordings of 138 subjects, 106 males and 32 females. The vocabulary consists of 56 two-digit numbers ranging from 21 to 97 pronounced as "twenty-one", "ninety-seven" and spoken continuously in sets of three, for example "36-45-89", in each utterance. There are four enrollment sessions per speaker, and each session contains 24 utterances. In the test set there are ten test sessions per speaker with 4 phrases per session. All waveforms are low-pass filtered at 3.8 kHz and sampled at 8 kHz. All the waveform files are compressed with a SPHERE header.

The Switchboard corpus is one of the largest collections of conversational, telephone speech recordings. There are two main Switchboard corpora (I and II), both were collected by a participant calling into an automated operator that connected him/her to another participant and recorded their conversation for 5 minutes. In Switchboard there are different noises include echo or crosstalk in the telephone circuit, background noise (e.g. baby crying, television, radio, etc.) and distortion (refers to echo and other recording problems). There are 543 and 657 speakers in Switchboard I and II corpora respectively. Different Speaker Recognition Evaluation (SRE) corpora were derived from Switchboard to allow assessment and comparison of different systems. The data used in this project, the 2002 and 2003 NIST SRE

subsets of Switchboard (Linguistic Data Consortium 2002; Linguistic Data Consortium 2003) were obtained through National Institute of Standards and Technology (NIST) and Linguistic Data Consortium (LDC) to enable us to evaluate the segmental GMM for speaker verification on the NIST 2003 SRE test.

2.4 Speech Dynamics and Speaker Recognition

Tosi et al. (1972) recognized the importance of slopes of formants during liquids, glides and diphthongs in speaker identification from visual examination of spectrograms. Stevens (1971) linked the variability in the slope of the formants for those sounds to "learned articulatory habits" of different speakers. In studying the acoustic features for speaker identification, Sambur (1975) also noticed that the slope of the formant was quite variable among speakers and "demonstrated excellent identification potential".

Yang et al. did some interesting experiments to investigate the relationship between the formant trajectories and the vocal tract model (1996). They used Fant's (1960) acoustic lossless tube model. The model configuration is defined by the parameters representing the physiological characteristics of the vocal tract of a speaker, and the parameters representing the properties of the speaker's voice patterns that reflect his learned behavior of speaking. They varied the two sets of parameters and examined the formant trajectories. Their study clearly showed that the dynamic aspects of speech signals

(e.g. the position and slope of the formant trajectories) are different for different speakers. The variations in the speech dynamics can be caused by the speaker's learned way of speaking, as well as the size and shape of the speaker's vocal tract.

Ainsworth (1996) described experiments in which listeners were played pairs of synthesized vowels. The synthesis method was formant synthesis based on two formants. The only difference between the two utterances in a given pair was the formant transition between the vowels. The listeners were requested to judge if the two synthesized vowels of each pair sounded the same. The results show that although there are different formant transitions between the sounds, to a degree they do not make a difference to the listeners' perception of the sounds. The formant transition comes from the dynamics of the vocal tract movement during the production of speech, which varies from speaker to speaker. If listeners can tolerate variation of the formant transitions then this may indicate the potential for the presence of differences between individuals in these dynamic regions.

2.5 Summary

This chapter presented background knowledge of speaker recognition. The project focuses on text-dependent and text-independent speaker verification. The key issue is to improve the performance of a speaker verification system by implementing the dynamic information which could contain individual differences. The role of speech dynamics in the speaker verification systems will be investigated and discussed through the experiments and studies.

As mentioned at the beginning of this chapter, the general approach to speaker recognition consists of several main steps: feature extraction, pattern matching, speaker reference model generation, and decision making. The detailed methods of these steps will be introduced in the following chapters.

CHAPTER 3

Front-End Analysis

Front-end analysis is the first stage of speaker recognition, whereby the input acoustic signal is converted to a sequence of acoustic feature vectors. Ideally the front-end analysis should preserve all the important information while not being sensitive to irrelevant acoustic variations. This chapter will present the theory and methods of extracting the cepstral feature vectors which have been shown empirically to be an effective representation of speech signals and are commonly used for speaker recognition. The features which represent the dynamics of the speech signals will also be introduced.

3.1 Cepstral feature vectors

Different parametric representations of the speech spectrum were examined for their effectiveness for speaker recognition and the cepstrum was found to be the most effective among all the parameters investigated (Atal 1974). To-date the mel-frequency cepstral coefficients (MFCCs) (Davis and Mer-

melstein 1980) are perhaps the most popular acoustic representation of the speech signal spectrum used in speaker verification, and are also widely used in automatic speech recognition (ASR). MFCCs are extracted by Fourier transform based analysis, followed by a set of perceptually scaled filters which computes a weighted sum of Fourier spectral components. The idea of the scaled filters is to obtain a non-linear frequency based spectrum, inspired by the human perceptual system because the human ear resolves frequencies non-linearly across the audio spectrum (Stevens et al. 1937). The process to generate MFCCs from a speech signal is shown in Figure 3.1.

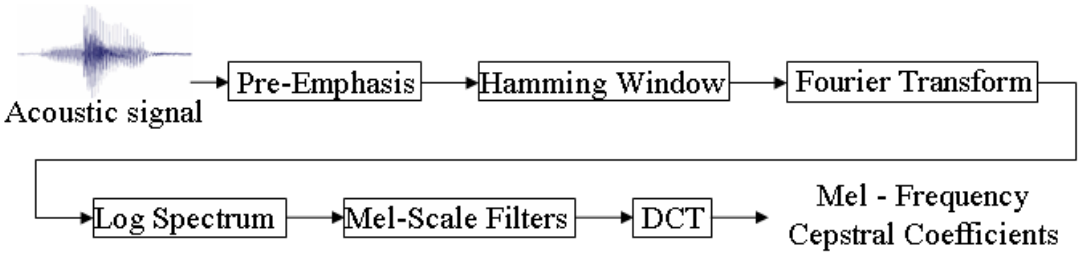


Figure 3.1: The Filterbank Analysis.

A pre-emphasis filter is first applied to the speech signal to compensate for a fall-off of energy with increased frequency in the upper part of the voiced speech spectrum. Psychophysical data on pure-tone thresholds suggests that a similar kind of high-frequency pre-emphasis occurs in the peripheral auditory system (Sivian and White 1933). The speech signal is then framed and each framed section is multiplied by a Hamming window (Harris 1978). Analyzing the speech signal into a sequence of frames achieves a reasonable approximation of the signal, in which each frame is represented by a single feature vector describing the average spectrum for a short time interval. The

application of the windowing is to reduce the possible discontinuities at the edges of the framed signal. The length of the window should be short enough to give the required time resolution and also should be long enough to provide adequate frequency resolution. For the duration covered by a single window, the speech signal is assumed as being stationary in terms of its spectrum. Commonly a 20-25 ms window is applied at 10 ms intervals, which gives a frame rate of 100 frames per second.

3.2 Mel-scale Filterbank Analysis

After applying a Fourier transform the magnitude spectrum is put through a bank of triangular filters, known as the mel-scale filterbank (Stevens et al. 1937), designed to match the critical bands of the ear. Hence perceptually important aspects of the short-term speech spectrum are captured. The triangular filters are distributed on a mel-frequency scale, which is approximately linear up to 1kHz and logarithmic above 1kHz. Each triangular filter has a unity magnitude at the center frequency. The magnitude decreases linearly at both sides to zero at the center frequencies of the two adjacent filters, as suggested in (Davis and Mermelstein 1980). The MFCCs are then obtained from the log filterbank amplitudes using a discrete cosine transform (DCT) to the output of mel filters. The first cepstral coefficient, mfc_0 is proportional to the mean of the framed signal and thus can be used as a measurement for energy. As j increases, the cepstral coefficient mfc_j captures increasingly fine detail of the spectral structure. MFCCs are commonly used in the stochastic modelling systems (e.g. GMM, HMM) which will be

introduced in Chapter 4. In a typical GMM speaker verification system up to 2048 Gaussian mixture components are contained in the model and approximately 38 MFCC parameters are extracted (Hansen et al. 2004). These include MFCC 1 to 18, the signal energy, and the first time derivative cepstral parameters, or the delta cepstral parameters. The delta cepstral parameters will be presented in Section 3.4.

An alternative to the Mel-scale filterbank analysis for representing the short-term spectrum is Linear Prediction (LP) analysis. The basic idea of linear predictive analysis is that a specific speech sample at the current time can be approximated as a linear combination of past speech samples. The N predictor coefficients can be thought of as the parameters of an N^{th} order all-pole linear filter. Perceptual Linear prediction (PLP) (Hermansky 1990) is one LP-based analysis method that successfully incorporates a non-linear frequency scale, which is very similar to MFCC analysis. Although not so popularly used in speaker recognition as MFCC, PLP and other LP-based analysis methods are frequently used in state-of-the-art speech recognition systems.

3.3 Spectral Variability Compensation

In practical applications the speech used to train and test a speaker recognition system may have been corrupted by either ambient noise or distortions from transmission over a communications channel. The feature vectors extracted from the distorted speech are also corrupted. As statistical modelling methods are based on modelling the underlying distribution of the feature

vectors, it is important to use some compensation methods on the feature vectors to remove the unwanted degradation and to improve the robustness of the speaker recognition systems. Commonly used methods include cepstral mean subtraction (CMS) (Atal 1974; Furui 1981), cepstral variance normalization (Viikki and Laurila 1998), noise masking (Klatt 1976), parallel model combination (PMC) (Gales and Young 1992), frequency warping (Reynolds 1992) and RASTA filtering (Hermansky and Morgan 1994). RASTA filtering will be discussed in 10.3.

Cepstral mean subtraction (Atal 1974; Furui 1981) is a normalization technique which calculates an average value of the cepstral coefficients of each channel and subtracts this average value from all coefficients in that channel. By applying CMS to the cepstral coefficients it is hoped to remove the channel effect which appears as convolutional in the time domain and hence becomes an additive constant in the log cepstral domain. This technique is very effective in practice, where it compensates for long-term spectral effects such as those caused by different microphones, telephone handsets and audio channels. CMS can be applied to each of the input speech files to remove the different channel effects. It can also be applied to variable lengths of speech to remove the varying channel effects within a file, such as the change of microphone positions. CMS has been widely used for both speech and speaker recognition systems.

Cepstral variance normalization (Viikki and Laurila 1998) is also commonly used. Given a cepstral vector and the calculated mean of the cepstral coefficients in each channel, the variance of the cepstral coefficients in each channel can be computed and used to normalize the coefficients in that

channel to scale the variance of the data to 1.0. This is to remove different cepstral coefficient distributions due to variable channel distortions. In this thesis cepstral mean subtraction and cepstral variance normalization are both used.

Noise masking (Klatt 1976) is based on the theory that the low-energy parts of the speech spectrum may be completely corrupted by the noise, but the higher-level parts of the spectrum above the noise level will remain unaffected. Noise masking uses a noise estimate to 'mask' both input speech and the models or templates used in the pattern matching methods.

Parallel model combination (Gales and Young 1992) is used to combine speech models with noise models in order to estimate models for the corrupted speech. Usually a noise model and a speech model are built on the acoustic domain. These models are then transferred to the linear spectral domain in which they can be combined because the model means are addable in the linear spectral domain. After combination the mixed 'noise-plus-speech' model is then transferred back to the acoustic domain.

Frequency warping (Reynolds 1992) is often applied to the magnitude DFT spectrum to avoid any differences in speech channel bandwidth. Basically the linear warping maps different frequency axes to a new frequency axis, to eliminate spectral components outside the specified frequency range and also to expand the spectrum to full bandwidth for subsequent processing.

3.4 Modelling speech dynamics

3.4.1 Dynamic features

The extracted feature vectors described above are broadly applied in different systems for speaker recognition and speech recognition. Among these systems statistical modelling methods, which will be described in the next Chapter, are most commonly used. Statistical modelling methods provide a framework which is broadly appropriate for modelling speech patterns. However, these models are simply general statistics pattern matchers, which do not consider the constraints inherent in the speech production process and make certain assumptions that are inappropriate for modeling speech patterns, including the piecewise stationarity, the temporal independence assumption and the geometric probability distribution for the state duration. The application of derivative parameters, or dynamic features, in speaker recognition systems using statistical modelling was motivated by the need to use the transitional spectral information to compensate for the piecewise stationarity assumption and the independence assumption of adjacent acoustic vectors (Furui 1981).

The delta parameters are the first-order time derivatives of the cepstral coefficients which are extracted at every frame period. For example, delta cepstra can be computed over ± 2 feature vectors (two to the left and two to the right over time) from the current vector (Soong and Rosenberg 1988). The second order derivatives, or the acceleration parameters, are in turn calculated using the delta parameters. It was shown that they successfully improve speaker recognition performance (Furui 1981; Soong and Rosenberg 1988). Subsequently these time derivatives were also used in speech recog-

nition. Because of the performance enhancement, acoustic vectors complemented by their first and second time derivatives are virtually always adopted in state-of-the-art ASR systems. Recently there is another technique called the Shifted Delta Cepstrum (SDC), which uses a development of simple delta cepstrum to substitute the delta and acceleration features in robust applications of speaker verification. The SDC has been found to exhibit superior performance to the delta and acceleration cepstra due to its ability to incorporate additional temporal information (Calvo et al. 2007). It is briefly introduced in 4.6.3.

3.4.2 Models which implement speech dynamics

In a conventional HMM, the assumptions that the underlying structure of a speech segment is stationary, and that the static, Δ and Δ^2 parameters are non-zero, are clearly inconsistent. A trajectory-based model could overcome this inconsistency: for such a model to incorporate non-zero Δ parameters, linear trajectories would be needed, while one which included non-zero Δ and Δ^2 would need quadratic trajectories. The issues raised by including dynamic features in a conventional HMM are discussed in (Bridle 2004). Alternative statistical modelling systems brought up new applications and understanding to implementing speech dynamics. Such models include trajectory HMMs (Tokuda et al. 2003) and segmental HMMs (Gales and Young 1993; Holmes and Russell 1999; Ostendorf et al. 1996).

Tokuda (Tokuda et al. 2003) uses another method, named trajectory HMM, to address this inconsistency. In Tokuda's research, an HMM whose

state observation vector includes static and dynamic parameters can be reformulated as a trajectory model by imposing the explicit relationship between the static and dynamic features. Basically, Tokuda's method derives a trajectory over a period of time which typically corresponds to multiple HMMs, which is more consistent with the sequence of HMM state static and dynamic parameters.

Stochastic segmental HMMs were introduced to address standard HMM's inappropriate assumptions. Segmental HMMs provide the opportunity to exploit acoustic information which is apparent at the segmental but not at the frame level. The segment refers to any sequence of frames representing some linguistically meaningful speech unit.The detailed theory of segmental HMMs will be presented in Chapter 5.

3.4.3 Other research on incorporation of dynamic information for speech recognition

There is other research on incorporation of dynamic information. Rather than to model dynamics directly, most of these methods use some type of filter to produce robust speech representations in noisy conditions. Although some of them have not yet been applied to speaker recognition experiments, they have been used in automatic speech recognition. I will talk about them briefly.

As mentioned earlier, the pattern-matching formalism based on HMM assumes that each acoustic observation vector is uncorrelated with its temporal neighbors. This assumption cannot be fulfilled by the transformed vectors

for the usual frame shifts which is typically 10ms. This is the reason that the delta and the acceleration features are normally included, being appended to the static vector. These two temporal sequences of differential vectors are computed by filtering the basic time sequence of spectral parameter vectors. CMS(Atal 1974; Furui 1981) and RASTA filtering (Hermansky and Morgan 1994) also filter each time sequence of spectral parameters to remove its dc and slowly variant components. By using linear filters, these methods help obtain more robust and more discriminative speech representations.

Nadeu used the term Temporal Filtering (TF) to cover all these techniques which use filtering to receive better speech representations (Nadeu et al. 1997). In automatic speech and speaker recognition, the signal is usually represented by a set of time sequences of spectral parameters (TSSPs) that model the temporal evolution of the spectral envelope frame-to-frame. These sequences are then filtered, using one or more of the TF techniques, either to make them more robust to environmental conditions or to compute dynamic features which enhance discrimination. Nadeu designed some temporal filters according to the settings of the systems, such as the number of features, the type of recognition task (speech recognition in his case), the noise characteristics, etc. These filters revealed a band equalization effect that emphasizes certain modulation frequency bands. His experimental results showed the use of properly filtered parameter sequences results in improved speech recognition performance for clean speech.

Nadeu et al. also came up with a method called the Frequency Filtering (FF) technique, which performs the equalization of the cepstral coefficients by filtering the frequency sequence of log filter bank energies (Nadeu et al. 1995).

Either a first-order or a second-order FIR filter can be used in FF. a usually used filter (Nadeu et al. 1995) has an impulse response $h(k) = 1, 0, -1$. Its transfer function is $H(z) = z - z^{-1}$. This filter is computationally simple, since for each band it only requires to subtract the log filter bank energies of the two adjacent bands. By applying this type of filter some dynamic information of the frequency can also be kept in the parameters. FF can be jointly applied with the TF in speech recognition and the combined approach can be used to design a robust set of filters (Nadue, Macho, and Hernando 2001).

Greenberg and Kingsbury developed a modulation spectrogram for speech recognition (Greenberg and Kingsbury 1997) which displays and encodes the speech signal in terms of the distribution of slow modulations across time and frequency. The modulation spectrogram represents modulation frequencies between 0 and 8 Hz, with a peak sensitivity at 4 Hz, corresponding closely to the long-term modulation spectrum of speech. To produce the modulation spectrogram, the speech signal, sampled at 8 Hz, is analyzed into approximately critical-band-wide channels via an FIR filter bank. Within each channel the signal envelope is processed and normalized, which are then analyzed by computing the FFT over a 250 ms Hamming window every 12.5 ms. The emphasis of modulations in the range of 0-8 Hz with peak sensitivity at 4 Hz acts as a matched filter that passes only signals with temporal dynamics characteristic of speech. The modulation spectrographic representation of speech is more stable than the conventional narrow-band spectrogram representation for both clean and noisy speech (Greenberg and Kingsbury 1997). In Greenberg and Kingsbury's study of speech recognition, the modulation

spectrogram perform worse than the PLP method on clean speech but outperform the PLP on noisy speech.

Milner and Vaseghi used a two dimensional cepstral-time features, which is called the Cepstral-Time Feature matrix, to overcome the temporal independence assumption of HMM (Vaseghi et al. 1993). The cepstral-time matrix is obtained from a two-dimensional DCT of a spectral-time matrix, one dimension is cepstrum and the other relates to time. The cepstral-time matrix was shown to improve the speech recognition performance in noisy conditions.

3.5 Summary

This chapter introduced the basic techniques used in front-end processing of the speech. The cepstral feature vectors were introduced as the most commonly used parametric representation of speech signals. Then the mel-scale filterbank analysis was presented. The mel-scale filterbanks are designed to capture the perceptually important spectral information. Mel-frequency cepstral coefficients are used as the parameterization of the speech signal in this thesis.

In practical applications the speech is usually not recorded under ideal circumstances. Different recording equipment such as microphones or handsets and the communications channels can distort the speech spectrum. Ambient noise can also lead to corrupted speech and hence affect the extracted cepstral features. Some compensation methods which could help remove unwanted distortion while keep important speech information were discussed.

Then the use of the derivative features was presented. The delta parameters and acceleration parameters are introduced to compensate for the assumptions of piecewise stationarity and temporal independence in the statistical modelling methods. Most of the benefits from derivative features are thought to be due to their ability to capture dynamic information.

Finally some other research on incorporation of dynamic information were briefly introduced. Although not used in speaker recognition, they have been applied in speech recognition and proved that the dynamic information can be beneficial to improve speech recognition performance. This also suggests that the dynamic information could be very useful to the speaker recognition systems.

CHAPTER 4

Stochastic Modelling for Speaker Verification

Stochastic modelling methods have so far been proved to be the most successful tools for speaker verification. Seeing the speech signal as a sequence of random vectors, stochastic modelling methods compute the likelihood of the sequence of vectors given the speaker model. By Bayes' rule the posterior probability of the speaker model given the observations can then be calculated. The parameters of the speaker models are estimated from a set of training speech data collected from the speakers.

In speaker verification, the posterior probability of the claimed speaker given the observation is calculated. The decision is usually made by deciding whether the degree of fit to the claimed speaker exceeds a threshold. If the score is bigger than the threshold, the person is recognized as the true speaker, otherwise the person is rejected as an impostor.

In this chapter the theory of stochastic modelling for speaker verification will be presented first. Then two main modelling methods and their probability calculation algorithms will be introduced: Gaussian mixture models for

text-independent speaker verification and hidden Markov models for text-dependent speaker verification. Then some important techniques used in speaker verification, such as score normalization and speaker adaptation, will be introduced. Finally, some state-of-the-art techniques which have been developed recently and successfully applied to speaker recognition experiments will also be introduced.

4.1 Theory of Stochastic Modelling

In speech recognition, the measure of the degree of match between a speech unit and some speech data is based on Bayes' rule. The speech unit could be a word or a phoneme. Suppose W is a particular word, Y is a set of feature observations which is extracted from the speech data using the front-end analysis described in Chapter 3. We wish to calculate the probability of the word W given the observations Y. Using Bayes' rule, the posteriori probability , $P(W\ Y)$, is calculated as:

$$P(W\ Y) = \frac{p(Y\ W)P(W)}{p(Y)}, \tag{4.1}$$

where $p(Y\ W)$ is the probability of Y given a model of the word W.

For speaker verification, however, we wish to measure the match between a particular speaker x and the observed feature vectors Y. If we put x, instead of W into (4.1), we have

$$P(x\ Y) = \frac{p(Y\ S_x)P(x)}{p(Y)}, \tag{4.2}$$

where $p(Y\ S_x)$ is the probability of Y given S_x, the model of speaker x.

Having calculated the probability of the speaker given the observations, what we should do then is to make a decision whether the speaker x is the authorized speaker or instead, an impostor. Choosing a value for the threshold can be very difficult because the calculated score may vary considerably from utterance to utterance, especially if there are changes in the environment or channel characteristics. In practice $P(x\ Y)$ is compared to $P(g\ Y)$, the probability of a "general speaker" (Carey et al. 1991) given the same observations, to normalize the speaker score. The probability of the general speaker g given Y is calculated as

$$P(g\ Y) = \frac{p(Y\ S_{UBM})P(g)}{p(Y)}, \tag{4.3}$$

where S_{UBM} is the model for the general speaker, which is usually called the "universal background model" (UBM).

Finally, we have the following formula for speaker verification system:

$$S(Y, x) = \frac{P(x\ Y)}{P(g\ Y)} = \frac{p(Y\ S_x)P(x)}{p(Y\ S_{UBM})P(g)}. \tag{4.4}$$

The background model S_{UBM} is trained using a large number of utterances obtained from a large population of speakers. The speaker model S_x, however, is trained using the utterances from the speaker whose identity is to be verified.

In the simplest form of the verification system, suppose we use phone-level hidden Markov models, then we have two models for each phone unit (Figure

4.1). The verification test score for each model is produced as a likelihood

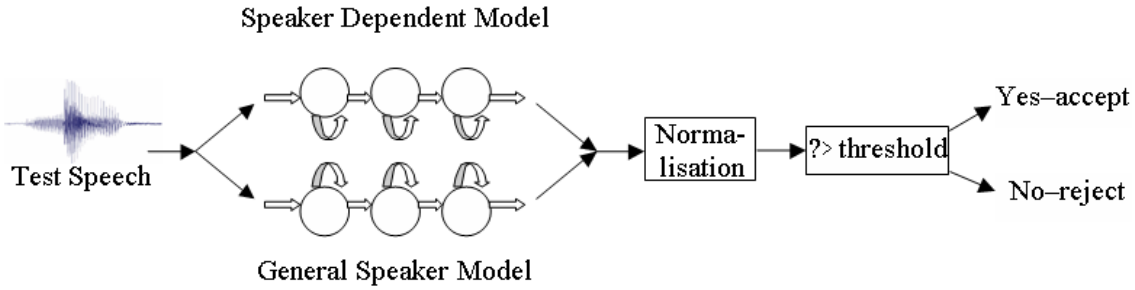


Figure 4.1: The Speaker Verification Process using an HMM system.

measure from a frame synchronous Viterbi search. The difference between the scores for the speaker dependent and background models is then computed and compared with a threshold. If the difference exceeds the threshold, then the person is accepted. Otherwise, the person is rejected as an impostor, as showed in Equations (4.5) and (4.6).

$$S(Y) = \frac{p(Y\ S_x)}{p(Y\ S_{UBM})} \frac{P(x)}{P(g)} > T_0, \tag{4.5}$$

or,

$$S(Y) = \frac{p(Y\ S_x)}{p(Y\ S_{UBM})} > T, \tag{4.6}$$

where T_0 and T are the thresholds.

From the perspective of measuring the performance of a speaker verification system, choosing a value for the acceptance threshold should also consider minimizing the number of verification errors. There are two verification errors: false acceptances, in which the system incorrectly accepts

an impostor, and false rejections, in which the system incorrectly rejects the true speaker. A high threshold reduces the number of false acceptances at the expense of more false rejections. Conversely, a low threshold reduces the number of false rejections but leads to more false acceptances. A measure of performance usually quoted for verification tasks is the Equal Error Rate (EER), which is the error rate obtained when the threshold is set so that the two types of error occur with equal probability.

There are two main methods, based on Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs), to model the speaker statistics and to represent the speaker identities. First introduced in 1990s, GMMs have become the dominant approach in text-independent speaker recognition applications over the past decade. A speaker dependent GMM is trained from unlabeled feature vector observations obtained from a given speaker, through a maximum likelihood procedure. The use of GMMs has demonstrated high speaker identification performance for 'content unknown' speech utterances. HMMs are commonly applied in text-dependent speaker verification. HMMs are always trained as a set of models each of which represents a phoneme or some other word or sub-word level unit spoken by a given speaker.

The following sections will introduce the theory, probability calculation and model parameter estimation of the two techniques. Some other techniques which have been developed recently (for example, the Support Vector Machines), are introduced in 4.6.

4.2 Gaussian Mixture Models

The Gaussian mixture model implements the Gaussian mixture density as a statistical model to represent speaker identities. A Gaussian mixture density, as showed in Figure 4.2, is a weighted sum of M component normal distributions, where each component distribution has a different mean and variance. Provided that there is a sufficient number of mixture components, any shape of distribution for a specific speaker can be approximated very closely by the combination of these Gaussian mixture components. Thus the

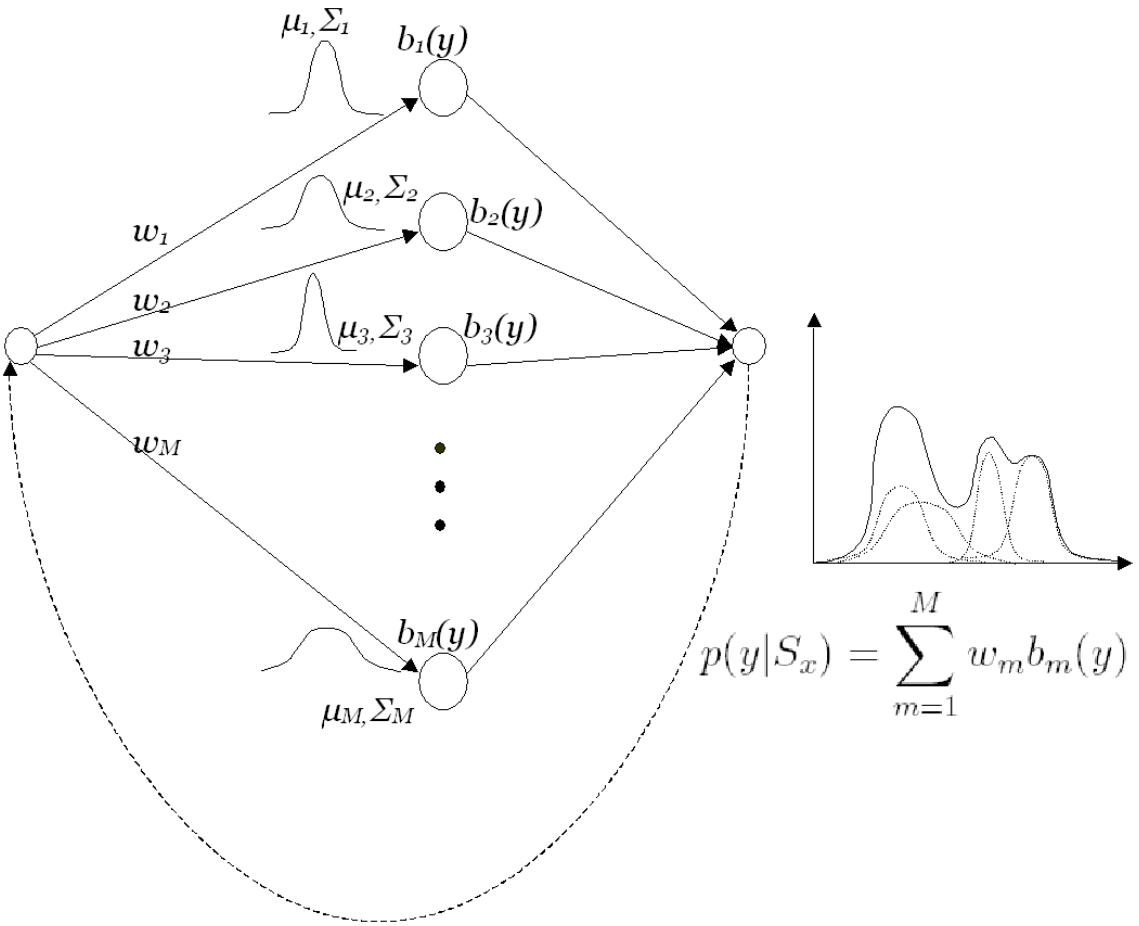


Figure 4.2: The Gaussian Mixture Model.

probability that a specific speaker model S_x generates an observation Y can be expressed as

$$p(Y\ S_x) = \sum_{m=1}^{M} w_m b_m(Y), \tag{4.7}$$

where Y is a D-dimensional observed feature vector[1], $b_m(Y), m = 1, 2, ..., M$, are the component normal densities, and $w_m, m = 1, 2, ..., M$, are the mixture weights, which satisfy the constraint that $\Sigma_{m=1}^{M} w_m = 1, w_m \geq 0$. If $\mathcal{N}(Y; \mu, \Sigma)$ is used to represent the probability density of the observed vector Y given a normal distribution with mean vector μ and covariance matrix Σ, the emission probability given by the m^{th} component of a Gaussian mixture density is

$$\begin{aligned} b_m(Y) &= \mathcal{N}(Y; \mu_m, \Sigma_m) \\ &= \frac{1}{(2\pi)^{D/2}\ \Sigma_m\ ^{1/2}} exp\left[-\frac{1}{2}(Y-\mu_m)^{\mathsf{T}}\Sigma_m^{-1}(Y-\mu_m)\right], \end{aligned} \tag{4.8}$$

where μ_m and Σ_m are the mean and covariance matrix associated with b_m, Σ_m is the determinant of Σ_m and $(.)^{\mathsf{T}}$ is the transpose operation. In the special case when the features are uncorrelated, the covariance matrix becomes a diagonal covariance matrix, having values of zero except along its main diagonal. Equation (4.8) can then be written as a product of probabilities given by the separate features:

$$b_m(Y) = \prod_{d=1}^{D} \frac{1}{\sigma_{md}\ \sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{y_d - \mu_{md}}{\sigma_{md}}\right)^2\right], \tag{4.9}$$

[1]For a clear view in Figure 4.2 the probability density of each component $b_m(Y)$ is shown as one-dimensional and the observation feature vector is drawn as a one-dimensional vector.

where y_d is the d^{th} feature of Y, and μ_{md} and σ_{md} are the mean and standard deviation of the distribution of the d^{th} feature for the m^{th} mixture component. Equation (4.9) is evidently computationally simpler than Equation (4.8). Most current speaker recognition systems assume the features to be uncorrelated and use diagonal covariance matrices.

For a sequence of observation vectors, $Y = [y_1, y_2, ..., y_T]$, assuming statistical independence between each vector, Equation (4.7) becomes

$$p(Y\ S_x) = \prod_{t=1}^{T} p(y_t\ S_x) = \prod_{t=1}^{T} \sum_{m=1}^{M} w_{xm} b_{xm}(y_t). \tag{4.10}$$

During speaker verification the probability of a test utterance, which consists of a sequence of observation vectors, conditioned on the speaker model S_x is calculated in this way and the log likelihood is derived.

$$\mathcal{L}p(Y\ S_x) = \sum_{t=1}^{T} logp(y_t\ S_x). \tag{4.11}$$

The log likelihood of the speaker model, subtracted by the log likelihood of the background model, is then compared with the preset threshold T and a speaker verification decision can be made.

$$\mathcal{L}p(Y\ S_x) - \mathcal{L}p(Y\ S_{UBM}) = \sum_{t=1}^{T} [logp(y_t\ S_x) - logp(y_t\ S_{UBM})] > T. \tag{4.12}$$

4.2.1 Parameter Estimation Methods

Maximum Likelihood Training

The general method for estimating the parameters of GMMs is Maximum Likelihood (ML) estimation. The maximum likelihood method estimates the stochastic model parameters to model the statistics of a set of observed samples so that the probability of the data conditioned on the models is maximized. Usually a training set containing a very large population of possible utterances is used for the estimation. The maximum likelihood training process can be formulated as determining the values of the model parameters in order to maximize the log likelihood of the training data.

Expectation Maximization Algorithm

Because we do not know which frames of training data corresponded to which model components, it is not straightforward to calculate a maximum-likelihood estimate of the parameters associated with each component. The expectation-maximization (EM) algorithm (Dempster et al. 1977) is a solution to this problem. Basically, if we have a set of rough estimates for all the model parameters, it is possible to compute new estimates for each parameter using the initial estimates, on the condition that the new estimates always produce a model that is at least as good as the old one in representing the data. If we iterate these operations a sufficiently large number of times the model will converge to a locally optimum solution. EM algorithm is now generally applied in GMM and HMM based stochastic modelling methods. A special case of EM algorithm, the Baum-Welch (BW) algorithm is commonly

applied in the HMM based modelling and will be introduced in 4.3.1.

Parameter Estimation Equations

Given a sequence of acoustic vectors $Y = [y_1, y_2, ..., y_T]$ for model training, and a GMM speaker model λ consisting of M Gaussian mixture components, we want to construct a new speaker model $\bar{\lambda}$ such that the new model $\bar{\lambda}$ increases the likelihood of the training data given the model, which can be described mathematically as

$$p(Y\ \bar{\lambda}) \geq p(Y\ \lambda). \tag{4.13}$$

As there is no direct solution, an auxiliary function of pairs of models, $Q(\lambda, \bar{\lambda})$, is defined to help find model $\bar{\lambda}$. It can be shown that if $Q(\lambda, \bar{\lambda}) \geq Q(\lambda, \lambda)$, then $p(Y\ \bar{\lambda}) \geq p(Y\ \lambda)$.

It turns out that Q has a unique critical point which is a maximum. Thus differentiating Q with respect to the elements of $\bar{\lambda}$, setting the results to 0 and solving gives a new set of model parameters $\bar{\lambda}$ so that $p(Y\ \bar{\lambda}) \geq p(Y\ \lambda)$. The reestimation formulae which define $\bar{\lambda}$ are in terms of the probabilities calculated with respect to the model λ.

The EM algorithm is an iterative process which begins with an initial estimate of the model parameters, λ_0. The process is applied repeatedly to obtain new models $\lambda_1, \lambda_2, ..., \lambda_n$ such that $p(Y\ \lambda_0) \leq p(Y\ \lambda_1) \leq ... \leq p(Y\ \lambda_n)$. When the difference between $p(Y\ \lambda_n)$ and $p(Y\ \lambda_{n-1})$ is so small, we can conclude that the process has reached a convergence.

The maximization of the auxiliary function with respect to each of the

parameters of $\bar{\lambda}$ ($\bar{w}_m, \bar{\mu}_m$ and $\bar{\Sigma}_m$) give the re-estimated equations for the component weights, the mean vector and covariance matrix as follows,

$$\bar{w}_m = \frac{1}{T}\sum_{t=1}^{T}\frac{w_m b_m(y_t)}{\sum_{k=1}^{M} w_k b_k(y_t)}, \tag{4.14}$$

$$\bar{\mu}_m = \frac{\sum_{t=1}^{T}\gamma_m(t) y_t}{\sum_{t=1}^{T}\gamma_m(t)}, \tag{4.15}$$

$$\bar{\Sigma}_m = \frac{\sum_{t=1}^{T}\gamma_m(t)(y_t - \bar{\mu}_m)(y_t - \bar{\mu}_m)^{\mathtt{T}}}{\sum_{t=1}^{T}\gamma_m(t)}. \tag{4.16}$$

where $\gamma_m(t)$ is defined to be the probability of being in component m at time t, and generating y_t, given that the model generates the whole sequence of T feature vectors Y.

$$\gamma_m(t) = P(Y\ \lambda)\frac{w_m b_m(y_t)}{\sum_{k=1}^{M} w_k b_k(y_t)}. \tag{4.17}$$

A model structure is built with a set of initial values for the parameters. These initial parameter values are then used to calculate re-estimated parameters, based on Equations (4.14) to (4.16). After iteratively estimating the model parameters can reach a locally optimum solution. The initial parameter values are very important as suitable starting estimates for the models can lead to a good local optimum. However, the chance of finding a global optimum is very small as the number of possible local optima is generally believed to be so vast.

4.3 Hidden Markov Models

Hidden Markov models were introduced into speech recognition in the 1970s (Neuberg 1971; Jelinek 1976; Tappert 1976). The basic methodology was invented in the early 1900s by A. A. Markov, a Russian mathematician. During the 1980s HMMs became the most popular speech recognition method (Poritz 1982; Rabiner et al. 1983; Juang 1984). At first researchers used discrete HMMs whose emission probability density function (pdf) is represented as a discrete distribution. Later continuous HMMs were brought forward whose emission pdf is represented as a parameterized continuous distribution. HMM based modelling is generally applied in most up-to-date text-dependent speaker verification systems.

In HMM based speech modelling, a Markov model is built as a symbolic model which represents a sub-word unit. As shown in figure 4.3, a markov model is a finite state machine which changes state once every time unit. At each time t that a state j is entered, the probability of a speech vector y_t being emitted can be calculated using the probability density function b_j which is associated with state j. For continuous HMMs the probability function is normally a GMM. The transition probability network within the model decides the transition from state i to state j, which is typically referred to as the discrete probability a_{ij}. Generally a left-right HMM is used in which j is set to be always no smaller than i if $a_{ij} > 0$. a_{ii} represents the "self-loop" probability on state i. E and F are the entry and final exit states. They are non-emitting null states for connecting models into sequences.

Suppose we have a vector sequence Y which corresponds to a known

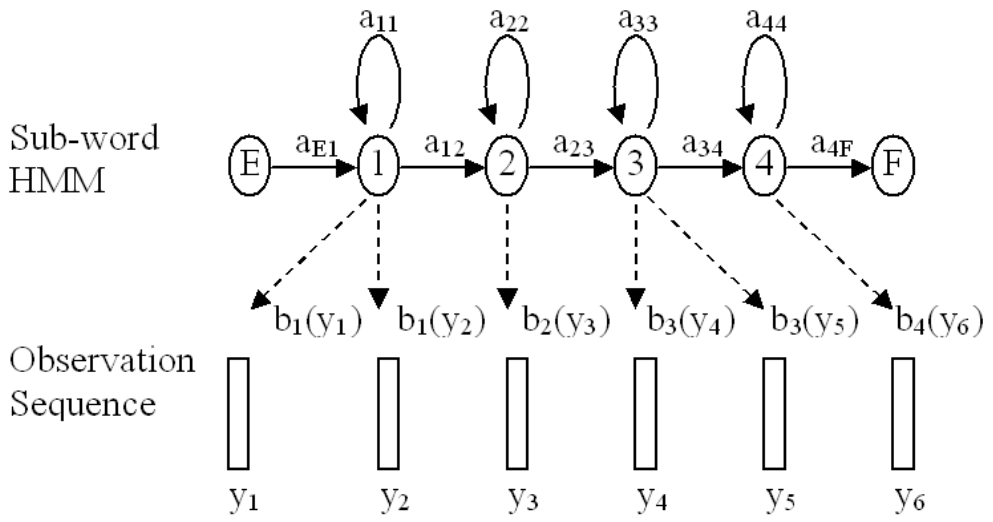


Figure 4.3: The Hidden Markov Model.

sequence of words, and λ is an HMM obtained by concatenating the necessary sequence of word- or phone-level HMMs. Thus, the joint probability of the vector sequence $Y = [y_1, y_2, ..., y_T]$ and a state sequence $X = x_1, x_2, ..., x_T$, given model λ is

$$p(Y, X\ \lambda) = a_{Ex_1}[\prod_{t=1}^{T-1} b_{x_t}(y_t)a_{x_t x_{t+1}}]b_{x_T}(y_T)a_{x_T F}, \tag{4.18}$$

The probability of Y is the sum of the probability $p(Y, X\ \lambda)$ over all possible state sequences X:

$$p(Y\ \lambda) = \sum_X p(Y, X\ \lambda). \tag{4.19}$$

As in GMM based modelling for speaker verification, the probability of a test utterance conditioned on the speaker model is calculated and the log

likelihood is derived. The log likelihood of Y given the speaker model, normalized with the log likelihood of Y given the background model, is then compared with the preset threshold and a speaker verification decision can be made.

The following Sections will present two main re-estimation algorithms for the HMM parameters, Baum-Welch algorithm and Viterbi algorithm.

4.3.1 Baum-Welch algorithm

The Baum-Welch (BW) algorithm is an EM algorithm which exploits forward and backward probabilities to re-estimate HMM model parameters. After re-estimation using a set of initial models and the BW algorithm, the probability of the training data given the new set of models is guaranteed to be higher than the probability for the previous model set, except at the critical point at which a local optimum has been reached. Thus the training procedure can be repeated until the difference between the new and old probabilities is sufficiently small, which indicates that the training process is close enough to its local optimum.

Forward and Backward Probabilities

Suppose that we have an utterance Y which corresponds to a model λ and comprises the sequence of feature vectors y_1 to y_T. The forward probability, $\alpha_j(t)$, is defined to be the probability of the model having emitted the first t

observed feature vectors, and that state j is occupied at time t:

$$\begin{aligned} \alpha_j(t) &= p(y_1, y_2, ..., y_t, s_t = j) \\ &= \left[\sum_{i=1}^{N} \alpha_i(t-1)a_{ij}\right] b_j(y_t) \qquad for\ 1 < t \leq T, \end{aligned} \tag{4.20}$$

where N is the total number of states. The backward probability, $\beta_j(t)$, is defined to be the probability of the model emitting the remaining $T - t$ observed vectors, given that the j^{th} state was occupied at frame t:

$$\begin{aligned} \beta_j(t) &= p(y_{t+1}, y_{t+2}, ..., y_T\ s_t = j) \\ &= \sum_{j=1}^{N} a_{ij} b_j(y_{t+1}) \beta_j(t+1) \qquad for\ T > t \geq 1, \end{aligned} \tag{4.21}$$

Combining the forward and backward probabilities, the probability of the model emitting the full set of T feature vectors given that the j^{th} state is occupied for the t^{th} observed vector is

$$p(y_1, y_2, ..., y_T, j_t) = \alpha_j(t)\beta_j(t). \tag{4.22}$$

As there are N possible states which could be occupied at time t, the probability $p(Y) = p(y_1, y_2, ..., y_T)$ can be calculated by

$$p(y_1, y_2, ..., y_T) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t) \qquad for\ any\ value\ of\ t. \tag{4.23}$$

In particular, choosing $t = T$, $p(y_1, y_2, ..., y_T) = \alpha_F(T)$.

Parameter Re-estimation

Similarly as in a GMM system, $\gamma_j(t)$ is defined to be the probability of being in state j at time t, given that model λ generates the whole sequence of T feature vectors Y. This term can be derived from $\alpha_j(t)\beta_j(t)$ using Bayes'

$$\begin{aligned}\gamma_j(t) &= p(j_t\ y_1, y_2, ..., y_T) = \frac{p(y_1, y_2, ..., y_T\ j_t)p(j_t)}{p(y_1, y_2, ..., y_T)} \\ &= \frac{p(y_1, y_2, ..., y_T, j_t)}{p(y_1, y_2, ..., y_T)} = \frac{\alpha_j(t)\beta_j(t)}{\alpha_F(T)}. \end{aligned} \tag{4.24}$$

$\alpha_F(T)$ is the value of the forward probability calculated at the last frame in the observation sequence. It is thus the probability of the complete set of observations being produced by the model. The normalization in Equation (4.24) by $\alpha_F(T)$ thus ensures that when there are several examples of the utterance for the model, all frames of all examples will contribute equally to the re-estimation.

Assuming there are E examples of the utterance, the re-estimates of the mean vector μ_j and the covariance matrix Σ_j of the emitting pdf associated with state j are given by:

$$\bar{\mu}_j = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)y_{te}}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)}, \tag{4.25}$$

$$\bar{\Sigma}_j = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)(y_{te}-\bar{\mu}_j)(y_{te}-\bar{\mu}_j)^{\mathsf{T}}}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)}, \tag{4.26}$$

where T_e denotes the number of frames for the e^{th} example and y_{te} is the feature vector at the t^{th} frame of the example e. $\gamma_j(t,e)$ represents the value of $\gamma_j(t)$ for the e^{th} example.

In order to re-estimate the transition probabilities, we need to define $\xi_{ij}(t)$ to be the probability that there is a transition from emitting state i to emitting state j at time t, and that the model generates the whole sequence of feature vectors corresponding to the sub-word unit:

$$\xi_{ij}(t) = \frac{\alpha_i(t)a_{ij}b_j(y_{t+1})\beta_j(t+1)}{\alpha_F(T)} \qquad for\ 1 \leq t < T. \tag{4.27}$$

The probability of a transition between any pair of states i and j is obtained by summing the values of $\xi_{ij}(t)$ over all frames for which the relevant transition is possible. Dividing this quantity by the total probability γ_i of occupying state i gives the re-estimate for a_{ij}:

$$\bar{a}_{ij} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e-1}\xi_{ij}(t,e)}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_i(t,e)} \qquad for\ 1 \leq i,j \geq N, \tag{4.28}$$

where $\xi_{ij}(t,e)$ denotes the value of $\xi_{ij}(t)$ for the e^{th} training example.

4.3.2 Viterbi Algorithm

The Viterbi algorithm is commonly used as a computationally less demanding alternative to the Baum-Welch algorithm (Viterbi 1967; Forney 1973). When calculating probabilities in the Baum-Welch algorithm, every possible state sequence has to be considered. In Viterbi algorithm, however, the calculations are substantially simplified by just considering the most likely state sequence.

$$\widehat{p}(y_1, y_2, ..., y_T) = max_X[p(y_1, y_2, ..., y_T, X)], X = x_1, x_2, ..., x_T. \tag{4.29}$$

The statistics for state j are therefore gathered over all examples of the phone-level unit using all frames for which state j is occupied. The re-estimate of the transition probability is given by:

$$\bar{a}_{ij} = \frac{n_{ij}}{n_i} \qquad for\ all\ pairs\ of\ emitting\ states,\ 1 \leq i, j \leq N, \tag{4.30}$$

where n_i is the number of frames for which the state i is occupied, and n_{ij} is the number of frames for which a transition occurs between state i and j.

If we define s_{te} to denote the state occupied at frame t for example e, the re-estimates of the mean vector and the covariance matrix are then given by:

$$\bar{\mu}_j = \frac{1}{n_j} \sum_{e=1}^{E} \sum_{s_{te}=j} y_{te} \tag{4.31}$$

$$\bar{\Sigma}_j = \frac{1}{n_j} \sum_{e=1}^{E} \sum_{s_{te}=j} (y_{te} - \bar{\mu}_j)(y_{te} - \bar{\mu}_j)^{\mathsf{T}}. \tag{4.32}$$

As with the Baum-Welch re-estimation, the Viterbi training procedure is applied in an iterative manner until the increase in the likelihood of the training data is arbitrarily small. Because the contribution to the total probability is usually much greater for the most likely path than for all other paths, Viterbi training usually gives similar models to those trained using the Baum-Welch. This optimization reduces computational load and additionally allows the recovery of the most likely state sequence. Therefore the Viterbi training is often adopted as an alternative to full Baum-Welch training. In many HMM system implementations, the Viterbi algorithm is also

used for evaluation at recognition stage.

4.4 Score Normalization

An important issue in the statistical approaches to speaker recognition is score normalization, which includes the scaling of likelihood scores, and handset normalization. The scaling of the likelihood score distributions of different speakers is used to find a global speaker independent threshold for the decision making process. Handset or channel normalization is used to reduce unwanted environmental effects on the verification decision.

There are several commonly used normalization techniques. One is based on the use of a speaker independent background model (Carey et al. 1991), which was already mentioned in 4.1. For a test utterance, instead of using its log-likelihood score given a speaker model for classification, the relative log-likelihood score between the speaker model and a speaker independent background model is calculated for recognition.

Another is cohort normalization (Rosenberg et al. 1991), which uses speech from a set of cohort speakers who are close to the target speaker to estimate the parameters of the background model. If all speakers are included in the cohort, the cohort model equals the background model. The selection of the cohort can be done during training or testing.

Test normalization (T-norm) (Auckenthaler et al. 2000) is a commonly used approach for speaker verification systems. It has shown significant improvement for speaker verification performance. Also a distribution scaling approach, T-norm uses an impostor cohort to calculate a mean and variance

for each utterance to scale the speaker scores. During verification, some randomly chosen impostor models are tested against a test utterance, to produce the impostor log-likelihood scores for that utterance. Then a mean and variance are estimated from these scores. These parameters are used to perform score normalization, given by:

$$S_N = \frac{S - \mu}{\sigma}, \tag{4.33}$$

where μ and σ are the estimated mean and variance of the impostor distribution for an utterance, S is the original log-likelihood score, and S_N is the final log-likelihood score after T-norm. During testing we want to make sure that the "cohort" doesn't contain the true speaker model, which is impossible. Instead we choose a big cohort so that even if the true speaker model is included in the cohort, it doesn't affect the mean and variance of the cohort. According to Auckenthaler and Carey's research, a cohort of 50 impostors is big enough and improves speaker verification performance significantly. A cohort size above 50 impostors leads to no significant improvement in performance. So, T-norm modifies the scores so that the impostor score distribution has zero mean and unit variance. This helps bring down the variation of distribution of the impostor scores so that a uniform threshold can be chosen for all testing scores.

Another technique used to reduce environmental effects is zero normalization (Z-norm) (Reynolds 1997). Z-norm (also known as handset normalization or H-norm) was originally proposed for joint handset and speaker normalization. The basic approach is to estimate from development data

handset-dependent biases and scales in the log-likelihood ratio scores and then remove these from scores during operation. In Z-norm a speaker model is tested against example impostor utterances and the log-likelihood scores are used to estimate a speaker specific mean and variance for the impostor distribution.

The background model is applied in this thesis. T-norm is applied in most experiments in this thesis. For T-norm, sometimes the gender-dependent impostor cohorts are chosen, which means in each cohort only the impostors who has the same gender as the test speaker are chosen.

4.5 Speaker Adaptation

In practical speaker verification the background model is typically trained using a large amount of speech data from various speakers. The speaker dependent models, however, usually do not have much data for training, and maximum likelihood estimates tend to be unreliable when the data are sparse. In the circumstance that the training data is limited, a speaker adaptation technique is applied to train the speaker models instead of full reestimation. During the adaptation process the background model and a small amount of training data from each speaker are needed. The parameters of the background model are adjusted to provide a better match to the speaker data and hence to obtain an improved model of the speaker.

Various speaker adaptation methods have been successfully applied to speaker recognition. The methods most often used are Maximum a Posteriori estimation (MAP) (Gauvain and Lee 1994), Maximum Likelihood Linear

Regression (MLLR) (Leggetter and Woodland 1995b), and the stochastic matching method (Sankar and Lee 1996). Of the above MAP and MLLR have been shown to improve the performance on speaker verification (Ahn et al. 2000).

4.5.1 MAP Adaptation

MAP adaptation, sometimes referred to as Bayesian estimation, incorporates prior knowledge about the model parameter distribution. The theory is that when there is only a limited quantity of training data, combining some prior information that we have about likely model parameter values with any available data should help in the model estimation. For MAP adaptation purposes, the generally used prior information is the background model, which is trained on a large amount of data from various speakers and is reliable. The acoustic vector space is also the parameter space for the state means of the speaker models. Hence the background model can be considered to be a PDF on the model parameter space, and hence is a possible prior. The new estimates are a weighted sum of the original model (background model) estimates and the observed data, with the relative contribution of the observed data depending on how much data is available. Research has shown that adapting only the mean leads to the best verification performance (Reynolds et al. 2000).

Suppose we have a trained HMM as the background model and some new observation data from a speaker, the MAP adaptation formula for state j is

$$\widehat{\mu}_j = r\bar{\mu}_j + (1-r)\mu_j, \tag{4.34}$$

where μ_j is the background model mean and $\bar{\mu}_j$ is the mean of the observed adaptation data from the speaker. The precise theoretically correct value of 'r' is given in (Gauvain and Lee 1994). However, it has been argued that the 'correctness' of this value is compromised by the assumptions which need to be made. A more pragmatic approach is to define r as follows, and thus is the approach taken in HTK.

$$r = \frac{N_j}{N_j + \tau}, \tag{4.35}$$

where τ is a weighting of the a prior knowledge to the adaptation speaker data. N_j is the sum of the probabilities that state j is occupied by each frame of the adaptation data. Thus if N_j is big, the new mean will be very close to the adaptation data. Otherwise the mean MAP estimates will remain close to the background model mean. With MAP adaptation, every single mean component in the system is updated with a MAP estimate, based on the prior mean, the weighting and the adaptation data. MAP adaptation is applied in this thesis to train speaker-dependent models.

4.5.2 Maximum likelihood linear regression

MLLR uses a set of regression-based linear transforms to tune the mean and variance parameters of an HMM or GMM system so that each state in the initial system is shifted to be more likely to generate the adaptation data. Given a model mean vector μ, a new mean $\widehat{\mu}$ is given by

$$\widehat{\mu} = A\mu + b \tag{4.36}$$

where A is a transformation matrix and b is a bias vector, both of which can be estimated given some speech data for adaptation. The variance transform can either be estimated separately from the mean transform, or alternatively the system can be constrained so that the same transformation matrix A is also applied to transform the covariance matrix.

For speaker adaptation, MLLR has been found to give worthwhile gains in recognition performance with limited adaptation data, and performance then improves as the quantity of data increases (Leggetter and Woodland 1995a). It has also been found to be useful for adaptation to changes in the environment.

Compared to MLLR, MAP adaptation requires more adaptation data to be effective. When larger amounts of adaptation data become available, MAP begins to perform better than MLLR with a global transform, due to the detailed update of each component rather than the pooled Gaussian transformation approach of MLLR. However, MLLR can be applied in a flexible manner depending on the amount of adaptation data. As more data becomes available, improved adaptation is possible by using multiple transforms, each of which is more specific and applied to certain groupings of Gaussian components (Leggetter and Woodland 1995a). MLLR uses a regression class tree to group the Gaussians in the model set so that the transformations can be chosen according to the amount and type of adaptation data.

4.6 Other Techniques for Speaker Verification

There are some other techniques which have been developed recently and applied in speaker verification experiments. Some have been proven to be successful. The following sections will introduce some of the techniques.

4.6.1 Support Vector Machines

In recent years a new methodology based on Support Vector Machines (SVMs) has proved to be an effective method for speaker recognition (Campbell 2002; Wan and Renals 2005). An SVM is a two-class classifier which makes it a natural solution to speaker or language recognition. SVMs perform a nonlinear mapping which transforms inputs into a high dimensional space and then separate classes with a hyperplane. During training the support vectors are obtained by an optimization process which relies upon a maximum margin concept. For a separable data set, the system places a hyperplane in a high dimensional space so that the hyperplane has maximum margin. SVMs have been combined with HMMs (Wan and Campbell 2000; Ganapathiraju and Picone 2000) and GMMs (Kharroubi et al. 2001).

The key design component in an SVM is the kernel, an inner product in the SVM feature space. Jaakkola and Haussler (Jaakkola and Haussler 1998) developed a Fisher kernel, which formed a link between generative (such as GMM or HMM) and discriminative models by using the Fisher score mapping. This technique maps a complete utterance onto a single point (in a

high dimensional space) using a generative model. Such a representation enables any suitable classifier to discriminate between complete utterances. Jaakkola and Haussler successfully applied it for biological sequence analysis. This method has then been combined with GMMs and applied to speaker recognition successfully (Fine, Navratil, and Gopinath 2001; Smith and Gales 2002; Wan and Renals 2003). Other score-space kernals include the Generalized Linear Discriminant Sequence kernel using polynomial vectors proposed by Campbell (Campbell 2002), and the Likelihood Ratio score-space kernel (Wan and Renals 2003).

Campbell et al. recently used the GMM supervector in a SVM classifier and proposed two SVM kernels based on distance metrics between GMM models: the GMM Supervector Linear kernel (Campbell et al. 2006) and the GMM L^2 Inner Product kernel (Campbell et al. 2006). The GMM supervector is a high-dimensional vector which is formed by stacking the means of the GMM model, which was trained using MAP adaptation on a given utterance. Thus the discrimination between utterances becomes the discrimination between the GMM supervectors. The GMM Supervector Linear kernel uses an approximation to calculate the KL divergence between two utterances. The GMM L^2 Inner Product kernel uses function space inner products to function a kernel.

4.6.2 Compensation Techniques

In addition to the score normalisation techniques already described, recent speaker and language identification systems employ sophisticated methods to

compensate for irrelevant, intersessional variability. In language identification this includes channel and speaker effects, and for speaker identification channel and other effects. As an example some of these methods exploit the GMM supervector representation (Campbell et al. 2006). Differences are computed between supervectors corresponding to the same class, and the resulting set of vectors is subject to PCA analysis for dimension reduction. The resulting low-dimensional characterization of intersession variability is used to normalize new data.

4.6.3 Shifted Delta Cepstrum

Another method, the Shifted Delta Cepstrum (SDC), was proposed by Bielefeld (Bielefeld 1994) and applied in language and speaker recognition (Torres-Carrasquillo et al. 2002; Allen et al. 2005; Kohler and Kennedy 2002; Campbell et al. 2006; Calvo et al. 2007). Typically, language and speaker recognition tasks use feature vectors containing cepstra and delta and sometimes acceleration cepstra. However, the SDC has been found to exhibit superior performance to the delta and acceleration cepstra due to its ability to incorporate additional temporal information, spanning multiple frames, into the vector.

SDCs are obtained by concatenating the delta cepstra computed across multiple frames of speech. The SDC features are specified by a set of 4 parameters, N, d, P and k, where N is the number of cepstral coefficients computed at each frame, d represents the time advance and delay for the delta computation, k is the number of blocks whose delta coefficients are

concatenated to form the final feature vector, and P is the time shift between consecutive blocks. Accordingly, $(k+1)N$ parameters are used for each SDC feature vector including the statics, as compared with $2N$ for conventional cepstra and delta cepstra feature vectors. For example, setting $N-d-P-k$ to $7-1-3-7$ (Torres-Carrasquillo et al. 2002; Campbell et al. 2006) results in a sequence of feature vectors of dimension 49 for each utterance.

The applications of the SDCs on the cepstral features for language identification with GMM (Torres-Carrasquillo et al. 2002) and SVM (Singer et al. 2003) have produced promising results. The applications of SDC in speaker verification (Calvo et al. 2007) show that SDC features become an alternative to MFCC, delta and acceleration features in robust applications of speaker verification, related to channel mismatch and session variability.

4.6.4 Exploiting High-level Information for Speaker Recognition

Current automatic speaker recognition systems have relied almost exclusively on low-level information via short-term features related to the speech spectrum. While these systems have produced very low error rates, they ignore other levels of information that convey speaker information, such as the particular word usage (idiolect), the pronunciation of the utterance, and the non-lexical utterances (sighs, laughs, hesitation sounds, etc.). Recently studies and works have been done to examine certain high-level information sources and have provided strong indications that potential gains are possible (Sonmez et al. 1998; Doddington 2001; Weber et al. 2002; Andrews et al.

2002).

In 2002 a SuperSID project for the exploitation of high-level information for high-performance speaker recognition was undertaken to develop new features and classifiers and to increase text-independent speaker recognition accuracy (Reynolds et al. 2003). In the SuperSID project the use of prosodic features such as the prosodic statistics and the dynamics of pitch and energy contours (Adami et al. 2003; Peskin et al. 2003) were examined. The phone N-grams (Andrews et al. 2002) or the phone binary trees (Navratil et al. 2003) were applied to use the time sequence of phones coming from a bank of open-loop phone recognizers to capture some information about speaker-dependent pronunciations. Similarly, the cross-stream phone modeling (Jin et al. 2003) and the pronunciation modeling (Klusacek et al. 2003) methods were applied to learn speaker-dependent pronunciations. For the lexical features, an n-gram idiolect system was implemented and used to examine the effects of using errorful word transcripts. They also examined the speaker information in turn-taking patterns and conversational styles, by using n-gram models of speaker turn characteristics (Peskin et al. 2003). Finally, by fusing these different levels of information, significant benefit can be gain even at extremely low error rates. This suggests that exploiting high-level information can help improve speaker recognition performance.

4.7 Summary

This Chapter presented the stochastic modelling methods which are typically used in state-of-the-art speaker recognition systems. GMMs have been

successfully applied in text-independent speaker recognition and HMMs have been generally applied in text-dependent speaker recognition.

The stochastic modelling methods see the speech signal as a sequence of independent random feature vectors. The model training and verification processes are based on computing the likelihood of a sequence of feature vectors given the model. Both systems use a maximum likelihood training algorithm and adaptation to train their models on the available training materials. The verification score is produced as the difference of log-likelihood measures from a Viterbi search.

The stochastic methods have some very desirable properties. They provide one framework within which the spectral characteristics (emission pdfs associated with states) and the temporal characteristics (transitions between states) are treated separately. This specialty is realized by a tractable mathematical framework for recognition and for training to match some given speech data. The model can be made to generalize to unseen data by the parameterized continuous distribution.

However, some assumptions are made in the stochastic model formalism that are clearly inappropriate for modelling speech patterns. Firstly, it is assumed that speech is produced by a piece-wise stationary process, with instantaneous transitions between stationary states. It is also assumed that the successive observations are independent. The model takes no account of the dynamic constraints of the physical system that has generated a particular sequence of acoustic data. The independence assumption is also the cause of the inappropriate geometric state duration distributions in HMMs as the probabilities for successive numbers of frames form a geometric progression.

To address the assumptions of independence and piece-wise stationarity, our approach is to associate individual states of models with variable-length sequences of acoustic feature vectors. With the segmental hidden Markov model it is possible to characterize both the duration of the segments and the relationship between the vectors in the sequence associated with the segment. The next Chapter presents the theory of segmental hidden Markov models.

CHAPTER 5

Segmental Hidden Markov Models

5.1 Motivation for Segmental HMMs

A "segmental HMM" (SHMM) can be defined in general terms as a Markov model where segments, rather than frames, are the homogeneous units which are treated as random variables associated with the model states. The idea of the "segments" was raised to associate models with variable-length sequences of acoustic feature vectors (Ostendorf et al. 1996). Some segment models introduces explicit state duration distributions to address the weak duration modeling of the HMMs (Russell and Moore 1985; Levinson 1986). Some segment models try to explicitly model correlation between observations, in which trajectories are usually incorporated to describe how the features change over time in the segment (Wellekens 1987; Brown 1987; Kenny et al. 1990; Russell 1993; Gales and Young 1993). With a trajectory segment model it is possible to capture both the duration of the segments and the relationship between the vectors in the sequence associated with each

segment.

A variety of trajectory segment models have been investigated, using different trajectories and different ways of describing the probability distributions associated with different trajectories. The different types of trajectory include constant (Russell 1993; Gales and Young 1993), linear (Russell 1993), linear dynamical systems (Digalakis 1992), exponential (Wiewiorka and Brookes 1996), 'smoothed piecewise constant' (Richards and Bridle 1999) and non-parametric (Ghitza and Sondhi 1993). A comprehensive review of segment models has been provided by Ostendorf, Digalakis and Kimball (Ostendorf et al. 1996).

5.2 Applying a linear trajectory SHMM to Speaker Verification

As the first step to apply segmental HMMs to speaker verification, a segmental HMM with a linear trajectory is applied in this thesis. It is a simple version of the "fixed-trajectory" segmental HMMs (Holmes and Russell 1999). Previous studies of trajectory representations of mel-cepstrum features (Holmes and Russell 1997) have suggested that a linear trajectory model is sufficient to capture the time-evolving characteristics. Gish and Ng (Gish and Ng 1993) also found that a linear trajectory was sufficient for most vowels in the vowel classification tests. Quadratic trajectory only benefit some diphthongs. The description of a more general model, the "Probabilistic-trajectory" segmental HMMs (PTSHMMs), can be found in Holmes and

Russell's paper (Holmes and Russell 1999).

Other studies were carried out using multiple-level segmental models to represent features in an articulatory domain (Russell and Jackson 2003; Russell and Jackson 2005). During training and recognition, features in the articulatory domain and observed features in the acoustic domain can be transformed into each other through one or more mapping functions. The motivation for such a model comes from the fact that in acoustic representations of speech (derived from short-term log-power spectra) articulator dynamics are manifested indirectly, often as movement between, rather than within, frequency bands. Therefore it would be better to model dynamics directly in an articulatory-based representation. The linear trajectory SHMM can be seen as a special case of the multiple-layer SHMM, with the articulatory-to-acoustic mapping set to an identity mapping.

For speaker verification, we wish to build a model containing the important information which can represent speaker characteristics. As mentioned in 2.4, during the production of speech, the formant transition comes from changes in the shape of the vocal tract during speech production, which may vary from speaker to speaker. Thus the dynamic spectral regions may indicate important differences between individuals. The segmental model should capture individual differences in non-stationary speech segments, which might otherwise be swamped by large variances due to the HMM piecewise stationarity assumption.

For example, we have some noisy data which are distributed around a trajectory, as shown in Figure 5.1. For convenience only one-dimensional data is shown here. As the conventional HMM system assumes both its

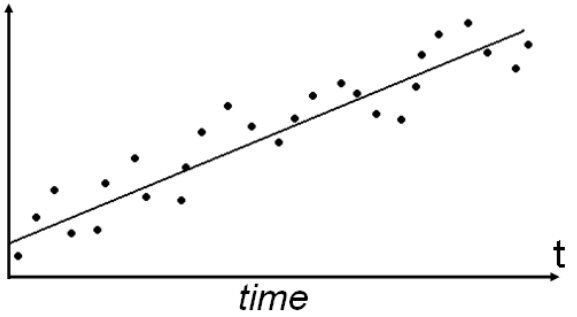


Figure 5.1: Illustration of the linear SHMM modelling assumption.

cepstral features and deltas to be stationary (Figure 5.2), this type of data is not desirable for the conventional HMMs. The values of both the features and the deltas vary too much and may well exceed the variation allowances of the model. However, for a segmental HMM system with linear trajectories (Figure 5.3), the data in Figure 5.1 may sit within the allowed variation range of the trajectory and hence can be modeled properly.

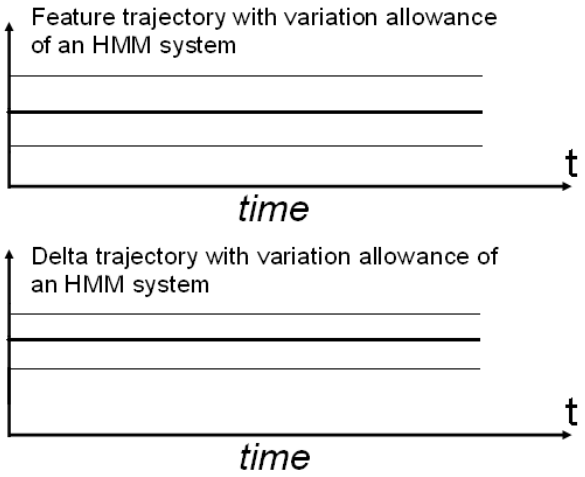


Figure 5.2: The Piecewise stationarity assumption of the HMM system.

As we can see from Figure 5.2 and Figure 5.3, the strategies of the two systems are differen. The conventional HMM system uses a mean and a

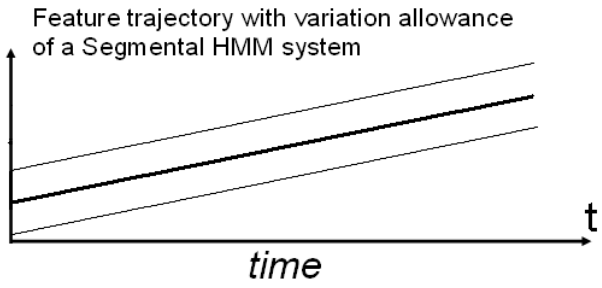


Figure 5.3: The dynamic trajectory structure of the Segmental HMM system.

variance to define its MFCC distribution. It also uses a mean and a variance to define its delta distribution. To assume both MFCCs and deltas are stationary is not appropriate. Only in the case that the data is stationary can the assumptions of the HMM system be fulfilled: the MFCCs are distributed around a constant and the deltas are distributed around zero. But this special case should not appear frequently in real time speech data. The SHMM system works differently. It uses a midpoint, a slope and a variance to define its MFCC distribution. If the data is stationary, the slope will be zero and the SHMM state is an equivalent to an HMM state. But if the data is not stationary, the SHMM will be able to catch any dynamics by using a nonzero-slope trajectory and allowing the MFCCs to be distributed around this trajectory.

It is plausible that such a model will improve our understanding of inter-speaker differences, and hence improve speaker recognition performance, by modelling some of the underlying mechanisms that give rise to intra- and inter-speaker differences.

Experiments have shown that segmental HMMs give better speech recognition results on TIMIT, demonstrating the benefits of incorporating the

segmental framework (Holmes and Russell 1996; Holmes and Russell 1997). We hope to see the improvement of performance appear in speaker verification experiments, for the speech dynamic information could be beneficial to speaker verification.

5.3 Segmental HMMs

A segmental HMM M is an N-state Markov model such that for each state $\sigma_i (i = 1, 2, ..., N)$ there exists a pdf b_i defined on the set of sequences of observation vectors Y. This pdf defines the probability that any segment is a valid instantiation of σ_i. To simplify notation, it is assumed that the state transition probability matrix A satisfies $a_{i,j} = 0$ if $j = i + 1$.

Let $Y = [y_1, y_2, ..., y_T]$ be an observation sequence and $X = [x_1, x_2, ..., x_T]$ ($x_t \quad \sigma_1, \sigma_2, ..., \sigma_N$) a state sequence, and let $t_{x,i}$ denote the time at which x enters σ_i. The joint probability of Y and x given M is

$$p(Y, x\ M) = \prod_{i=1}^{N} b_i(y_{t_{x,i}}, y_{t_{x,i}+1}, ..., y_{t_{x,i+1}-1}). \tag{5.1}$$

The probability $p(Y\ M)$ of Y conditioned on M is given by

$$p(Y\ M) = \sum_{x} p(Y, x\ M) \tag{5.2}$$

and is computed using an extended version of the HMM Baum-Welch algorithm (Holmes and Russell 1999).

5.4 Linear Trajectory SHMMs

Linear trajectory SHMM is a special form of the general Probabilistic-Trajectory SHMM (PTSHMM) (Holmes and Russell 1999), in which a model state is associated with a "probabilistic trajectory". A PTSHMM for a speech sound provides a representation of the range of possible underlying trajectories for that sound, where the trajectories are of variable duration and each duration has a state-dependent probability. In the PTSHMM there are two types of variation: the extra-segmental variation of plausible trajectories for any segment, and intra-segmental variation of the observations around any one trajectory.

5.4.1 Model Theory

In linear trajectory SHMMs a state treats an acoustic speech segment as a variable-duration, noisy function of a linear trajectory. A segment has different linear trajectories each of which represents one dimension of the feature vectors. Each trajectory has a mid-point mean value and a slope to specify how the acoustic features change over time (Figure 5.4). Each segment also has a duration probability to define the probability of segment length between one frame ($10ms$) and the maximum duration τ_{max}. The duration probability mass functions d_i were non-parametric (Ferguson 1980) in these experiments. During training the segmental Viterbi algorithm is used to segment the training data into state-level segments. The different durations of the segment samples in the data are counted. Then the probability of each possible segment duration is calculated by dividing each duration count with

the total number of samples.

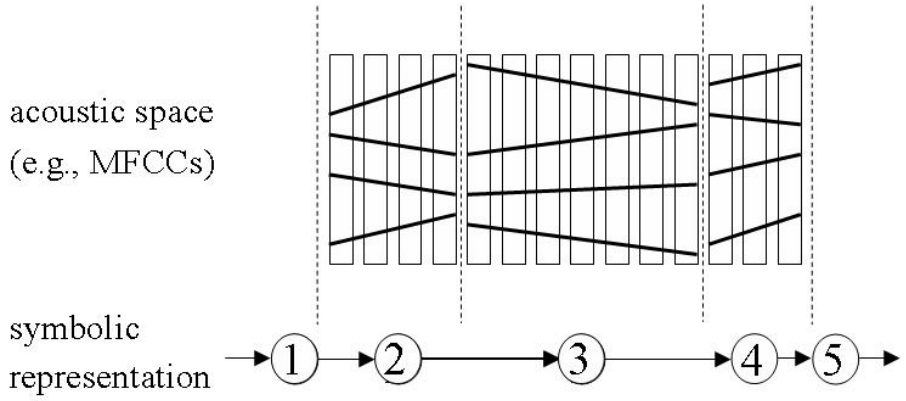


Figure 5.4: A segmental HMM that uses linear trajectories and durations to represent acoustic segments.

A state $\sigma_i (i = 1, 2, ..., N)$ is identified with a variable duration linear trajectory representing a speech signal in a D dimensional acoustic space, which, in our experiments, is based on MFCCs. Thus σ_i is parameterized by the mid-point vector $\mathbf{c}_i$ and slope vector $\mathbf{m}_i$, and a $D \times D$ covariance matrix V_i. A trajectory $\mathbf{f}_i$ of length T is defined by:

$$\mathbf{f}_i(t) = (t - \bar{t})\mathbf{m}_i + \mathbf{c}_i \tag{5.3}$$

where $\bar{t} = T/2$.

5.4.2 Model Parameter Estimation

To look at the model parameter estimation for the linear trajectory SHMMs, we should start from the general PTSHMMs. For reasons of mathematical tractability and of trainability, all variability in PTSHMMs is modelled with Gaussian distributions assuming diagonal covariance matrices. In a

PTSHMM the "extra segmental" variation for a state is defined by a PDF defined in the set of possible state trajectories. In (Holmes and Russell 1999) it is assumed that this PDF is Gaussian. In PTSHMMs we define the expected trajectory mid point mean of state σ_i as ν_i, the mid point variance as η_i, the trajectory slope mean as μ_i and the slope variance as γ_i. Suppose we have a single segment of feature vectors $Y = [y_1, y_2, ..., y_T]$, and the trajectory f has a mid point c and a slope m, the probability of the observation Y and trajectory f given state σ_i is

$$p(Y, f\ \sigma_i) = d_{\sigma_i}(T) p_{\sigma_i}(f) \prod_{t=1}^{T} p(y_t\ f(t)), \tag{5.4}$$

where d_σ is the duration PDF of state σ. So,

$$p(Y\ \sigma_i) = \int_f p(Y, f\ \sigma_i) df. \tag{5.5}$$

In a PTSHMM there is one intra-segmental probability per frame in a segment but also one extra-segmental probability per segment. Different explanations of the data may use different numbers of the two types of probability, depending on the number of segments. Recognition performance is thus dependent on a suitable balance between the different numbers of probability contributions, which compromises performance (Holmes and Russell 1999).

Two approximations to 5.5 are considered in (Holmes and Russell 1999). One is the "optimal trajectory" approximation, which was proposed by Russell (Russell 1993). This method is to use an approximation by considering

$p(y, f)$ for only one specified trajectory $\hat{f}$:

$$\hat{f} = argmax_f p(Y, f\ \sigma_i), \tag{5.6}$$

where $\hat{f}$ is the most likely trajectory. The probability of the observation sequence given state i hence can be written as:

$$p(Y\ \sigma_i) \cong p(Y, \hat{f}\ \sigma_i). \tag{5.7}$$

The second alternative method is the "fixed linear trajectory" SHMM. In the "fixed linear trajectory" SHMM the means of the PDFs which describe the trajectory distribution in a PTSHMM are treated as the actual mid-point and slope values of a single fixed linear trajecotry. Using this method, the probability of the observation sequence given state i is given by:

$$p(Y\ \sigma_i) \cong p(Y, \bar{f}\ \sigma_i), \tag{5.8}$$

where $\bar{f}$ is the linear trajectory with mid point m_i and slope c_i.

Holmes and Russell's research (Holmes and Russell 1999) shows that the "optimal trajectory" approximation method to PTSHMM doesn't work very reliably and the speech recognition performance is poor using this method. This appears to be because of the imbalance between the two types of probability, the intra- and extra- segmental probabilities. The speech recognition performance for the "fixed linear trajectory" SHMM, however, is almost as good as the PTSHMM in which $p(Y)$ is calculated properly using the integral. Thus in SEGVit, the software we developed to train and test our

segmental HMM system, the "fixed linear trajectory" method is chosen. Another reason to choose the "fixed linear trajectory" SHMM is because that the SEGVit software was designed to support a more complicated segmental HMM structure. This is called a multiple-layer segmental HMM (Russell and Jackson 2005). This multiple-layer model has an 'articulatory' intermediate layer which can be transferred into or from the acoustic domain using linear or non-linear mappings. In a single layer model, this integral in 5.5 is tractable because the PDFs are assumed to be Gaussian. This is also likely to be the case for a multiple-level model in which the "articulatory-to-acoustic" mapping is linear. However, for a non-linear mapping this is no longer the case. It was decided that for simplicity the "fixed linear trajectory" SHMM would be implemented in SEGVit. The "fixed linear trajectory" SHMM is the type of segmental model which is used in this thesis. In the following chapters the term SHMM actually means the "fixed linear trajectory" SHMM.

If $Y^T = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T]$ is a sequence of acoustic feature vectors and a sample for state σ_i, the probability density of Y^T given state σ_i is given by:

$$p(Y^T\ \sigma_i) = b_i(Y^T) = d_i(T) \prod_{t=1}^{T} \mathcal{N}\left(\mathbf{y}_t; \mathbf{f}_i(t), V_i\right), \tag{5.9}$$

where $d_i(T)$ is the probability that state σ_i emits a segment of length T, and $\mathcal{N}\left(\mathbf{y}_t; \mathbf{f}_i(t), V_i\right)$ is a D dimensional Gaussian probability density function with mean $\mathbf{f}_i(t)$ and covariance matrix V_i (it is assumed that V_i is diagonal). The case $\mathbf{m}_i = 0$ corresponds to a constant trajectory SHMM. If, in addition, d_i is a geometric probability density function then this is functionally identical to a conventional HMM except for an upper bound τ_{max} on state duration.

With a trajectory structure in the model, the probability calculations in a SHMM system take more computer running time than the calculations in a conventional HMM system. To save the computational load, in this thesis the SHMM model parameters are optimized using an estimation-maximization scheme based on segmental Viterbi decoding. $\hat{\alpha}_t(i)$ is defined to be the joint probability of the acoustic sequence $y_1^t = [y_1, y_2, ..., y_t]$ and the partial state sequence $X_1^t = [x_1, x_2, ..., x_t]$ which maximizes the probability $p(y_1^t, x_1^t)$ given the model parameters and given that the state at time $t+1$ is not state i. It can be written as:

$$\hat{\alpha}_t(i) = max_d max_j \hat{\alpha}_{t-d}(j) a_{ij} b_i(y_{t-d+1}, y_{t-d+2}, ..., y_t), \tag{5.10}$$

where d is the duration, a_{ji} is the transition probability from state i to state j, and $b_i(y_{t-d+1}, y_{t-d+2}, ..., y_t)$ is the state probability of the observation sequence of duration d, emitted by state i.

Because $\hat{\alpha}_t(i)$ is decided by the maximal value of duration d and the maximal value of $\hat{\alpha}_t(j)$ at state j, the Viterbi algorithm helps the probability to be traced back to find an optimal state sequence. Once this is finished, the slope $m'(Y^T)$ and mid-point value $c'(Y^T)$ of the linear trajectory which provides the best fit to Y^T (in a least-squared error sense) can be calculated. They are given by

$$\mathbf{m}_i'(Y^T) = \frac{\sum_{t=1}^{T}(t-\bar{t})y_t}{\sum_{t=1}^{T}(t-\bar{t})^2} \tag{5.11}$$

and

$$\mathbf{c}_i'(Y^T) = \overline{Y^T} = \frac{\sum_{t=1}^{T} y_t}{T}. \tag{5.12}$$

The segmental Viterbi training procedure is applied in an iterative manner until the system converges and a local optimum is reached.

5.5 Summary

This Chapter has introduced the segmental HMM as a new model for speaker verification. The SHMM has a better structure than the conventional HMM which makes it well suited for speaker modelling. The model associates its states with variable-length sequences of acoustic feature vectors. By modelling the dynamic regions in speech, which may reflect some of the underlying mechanisms that give rise to individual differences, this model better characterizes the variations of a person's voice and thus should improve speaker verification accuracy.

The theory of the original Probabilistic-Trajectory SHMM was presented as well as its two approximation methods. The "fixed linear trajectory" SHMM is an alternative which works about as same as the original PT-SHMM in terms of the speech recognition performance. The "fixed linear trajectory" SHMM is hence applied to text-dependent and text-independent speaker verification in this thesis. For model parameter estimation, the segmental Viterbi decoder provides an iterative maximum likelihood estimation technique.

Our application of SHMMs focused on text-dependent speaker verification on the YOHO corpus and text-independent speaker verification on the Switchboard speech database. The next two chapters examines many issues related to the training of speaker models and the performance of the SHMM

speaker verification system.

CHAPTER 6

Text-Dependent Speaker Verification

The most straightforward application of SHMMs to speaker recognition is text-dependent speaker verification (TD-SV). This is because a conventional TD-SV system typically uses phone-level HMMs, which can simply be replaced by the corresponding phone-level SHMMs.

Suppose that a sequence of acoustic feature vectors $Y = [y_1, ..., y_\tau]$ is claimed to result from subject S speaking a text W. The decision whether to accept or reject this claim is based on the likelihood ratio:

$$L(S) = \frac{p(Y\ S, W)}{p(Y\ W)} \tag{6.1}$$

where $p(Y\ S, W)$ is computed using a set of phone-level models for speaker S, configured to represent the text W, and $p(Y\ W)$ is calculated using a set of speaker-independent background models configured to represent W. If the likelihood ratio $L(S)$ is bigger than a preset threshold, the claim is accepted.

We built a conventional HMM system and a segmental HMM system,

both using the same set of context-sensitive triphone model labels. Both systems were trained on the TIMIT and YOHO training material and tested on the YOHO test set.

6.1 Experimental Method

6.1.1 Acoustic Parameterization

Our experiments used the YOHO (Higgins 1990) and TIMIT (Garofolo et al. 1993) speech corpora. An overview of both corpora can be found in 2.3. All the models were initialized using the whole TIMIT training set and were further trained on YOHO. The TIMIT and YOHO data were parameterized, using HTK (25 ms window, 10 ms fixed frame rate), into 13 dimensional feature vectors comprising MFCCs 1 to 12 plus energy.

No Δ or Δ^2 parameters were used. We have not yet used Δ or Δ^2 parameters in any of our previous SHMM based experiments. This is mainly because part of the motivation for the development of SHMMs is to obtain a better model of speech dynamics and thereby obviate the need for these parameters, and also to reduce the SHMM computational load.

6.1.2 Construction of initial acoustic models using TIMIT

For the first step of model building, matching monophone model sets of HMMs and SHMMs were constructed on the TIMIT training data. For both HMMs and SHMMs, each monophone model contains three left-to-

right emitting states and two non-emitting states (null states) at each end. The 'self-loop' state-transition probabilities were set to zero in the case of SHMMs, but were non-zero for the conventional HMMs. For the SHMMs, the emitting state uses trajectories to represent a segment of observations whose duration is τ. The maximum segment duration τ_{max} was set to 15 ($150ms$) and the duration probability mass functions $d_i(i = 1, 2, 3)$ were the non-parametric Ferguson duration model (Ferguson 1980). The structure of both models are shown in Figure 6.1.

The conventional HMMs were constructed using the Hidden Markov Model Tool Kit (HTK) (Young et al. 1997), and the SHMMs using the 'SEGVit' software developed at the University of Birmingham. In terms of the training scheme, the HMMs and SHMMs were trained using Baum-Welch (HTK) and Viterbi-based (SEGVit) training respectively. The states of the HMMs were associated with single Gaussian densities. This was for compatibility with the SHMM system, which currently cannot accommodate multiple-component Gaussian mixture densities. The monophone HMMs were initialized and reestimated using the HTK tools 'HInit' and 'HRest' respectively (Young et al. 1997). These monophone HMMs were also used to seed the monophone SHMMs, by setting the SHMM state mean and variance vectors equal to the corresponding HMM state mean and variance vectors, and setting the state slope vectors equal to zero.

The reestimated monophone models were then used to seed a set of context-sensitive triphone models. These models are phone models, conditioned by their preceding and following phonemes. For example, "th-ih+s" is the phoneme "ih" which follows a "th" and precedes an "s". Again the

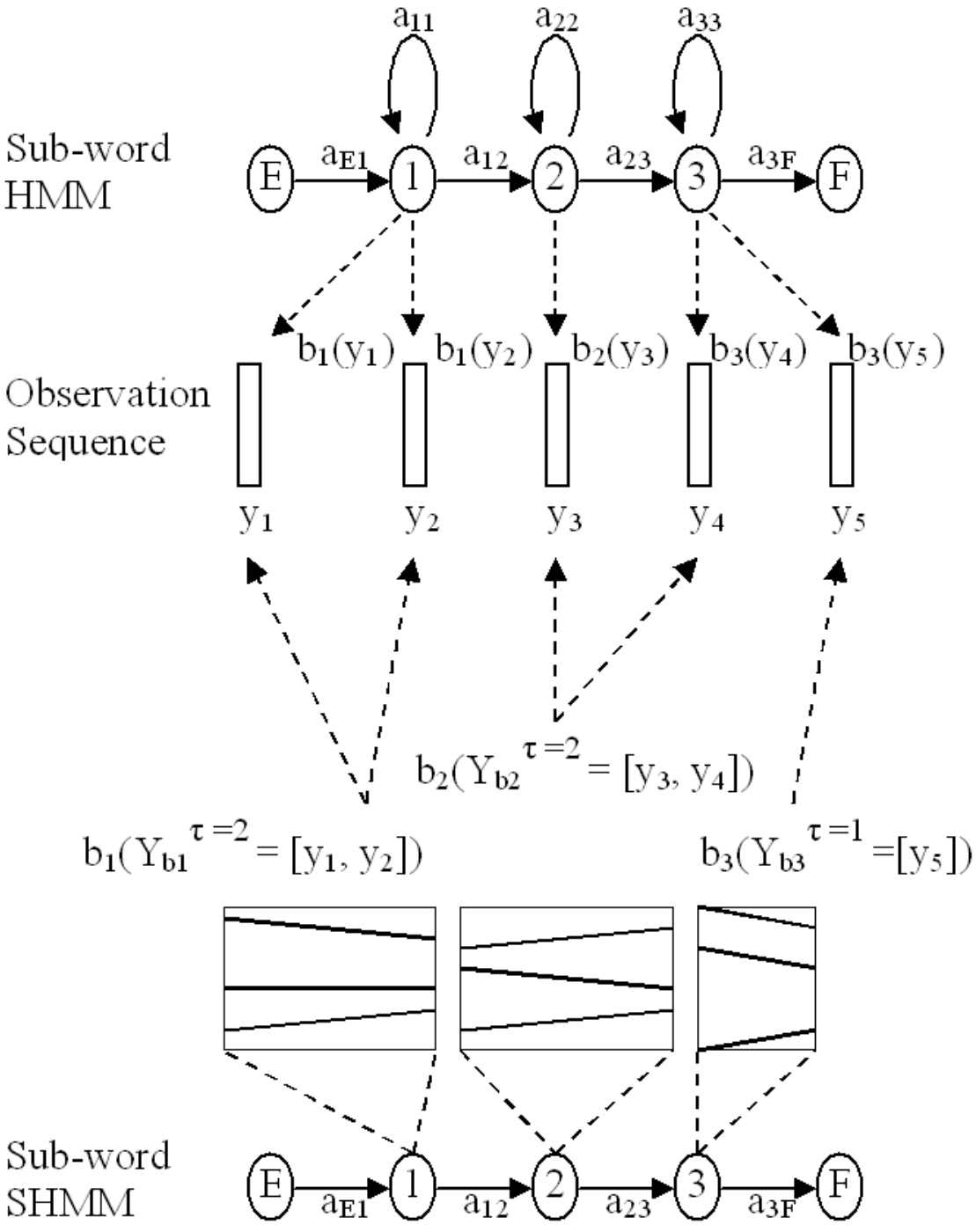


Figure 6.1: An HMM and a matching SHMM sub-word models.

TIMIT training data set was used to estimate the parameters for triphone HMMs and SHMMs. As SEGVit cannot perform state-level tying, the triphone model set was defined using a simple 'back off' procedure whereby a triphone model was constructed if 30 or more examples of that triphone context occurred in the training data, otherwise the triphone was replaced by a biphone which has only one context condition (if 30 or more examples of the biphone context occurred in the training data) or a monophone. This procedure forms the same 1400 model set that was used in (Russell and Jackson 2003). As some of the triphones from the YOHO triphone set do not appear to be in the TIMIT triphone set, 46 triphones in the 1400 TIMIT triphone set were used to model the 102 cross-word triphones in the YOHO corpus.

6.1.3 Model Training Using YOHO

Models for those triphones which occur in the YOHO data were used to seed speaker-independent YOHO HMMs and SHMMs. In our experiments we randomly chose 20 of the subjects (10 males and 10 females) in the YOHO enrollment set to train the speaker-independent HMMs and SHMMs. All of the data from the 20 speakers contain 1920 utterances. The YOHO vocabulary consists of 56 two-digit numbers ranging from 21 to 97 pronounced as "twenty-one", "ninety-seven" and spoken continuously in sets of three, for example "36-45-89", in each utterance. The training data from these 20 subjects are excluded from speaker-dependent HMMs and SHMMs training. The models trained on the 20 subjects' material formed the HMM and SHMM background models. The HMM and SHMM UBMs were each trained using

20 iterations of Baum-Welch and Viterbi-based training respectively.

The remaining 118 subjects in the YOHO enrollment set were used as test subjects. For each of these subjects, 96 utterances were used to train the HMM and SHMM speaker-dependent models (SDMs). As with the UBMs, the HMM and SHMM SDMs were trained using 20 iterations of Baum-Welch and Viterbi-based training, respectively.

6.1.4 Speaker Verification

In the YOHO test set, the 20 speech files for each of the 118 test subjects were split into 5 test sessions, each containing 4 speech files. A single speaker verification experiment consisted of comparing one such test session with a SDM and UBM. Thus, for each system, the number of 'authorised user' trials is $118 \times 5 = 590$, and the number of 'impostor' experiments is $118 \times 117 \times 5 = 69030$.

6.2 Results of text-dependent speaker verification experiments on YOHO

The results of the text-dependent speaker verification experiments are shown as Detection Error Tradeoff (DET) curves in Figure 6.2. The DET curve (Martin et al. 1997) is commonly used in speaker verification as a way to represent the system performances where trade offs of two types of errors are involved. The false alarm probability, or the false acceptance (incorrectly accepting an impostor) rate, is plotted on the horizontal axis, while the miss

probability, or the false rejection (incorrectly rejecting the target speaker) rate, is plotted on the vertical axis. Generally speaking, the closer the whole curve is to the origin, the better the system performance is. On each DET curve an optimal point can be marked out at which the trade off between the two types of error rate is optimal, depends on what purpose the speaker verification system is built for. The Equal Error Rate (EER) is also referred to compare system performances. It is the error rate obtained when the threshold is set so that the two types of error occur with equal probability.

For the DET curves of both our HMM and SHMM systems in Figure 6.2, the lower-bound of 0.17% for the false rejection probability equates to a single rejection out of the 590 authorised user trials. It is likely that this results from incorrectly labelled data.

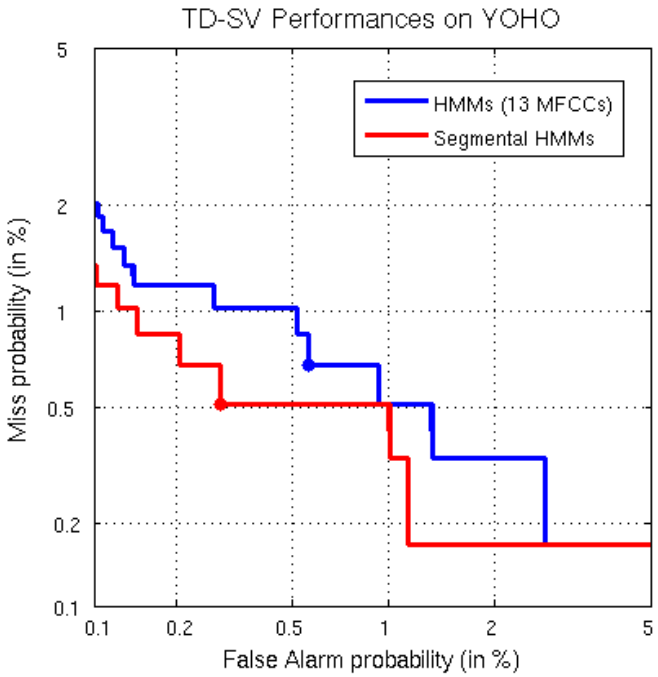


Figure 6.2: TD-SV results on YOHO using HMMs (dashed line) and SHMMs (solid line).

The results show that the SHMM system outperformed the conventional HMM system. The false rejection rates for the HMM and SHMM systems at the optimal points are 0.68% and 0.51%, corresponding to 4 and 3 false rejections respectively out of the 590 authorised user trials. This equates to a 25% reduction in the number of false rejections by using the SHMM system. The false acceptance rates for the HMM and SHMM systems at the optimal points are 0.52% and 0.29%, corresponding to 359 and 200 false acceptances respectively out of the 69,030 impostor trials. This equates to a 44% reduction in the number of false acceptances by using the SHMM system, relative to the conventional HMM-based system.

We didn't use the Δ or Δ^2 parameters in our HMM and SHMM models. This is partly because the goal is to assess the utility of dynamics, not to achieve overall optimal performance, and also to reduce the SHMM computational load. The motivation for the development of SHMMs is to obtain a better model of speech dynamics and thereby obviate the need for these parameters. A side effect of this is that the number of parameters in the SHMM system is greater than the conventional HMM system. So the SHMM system has an unfair advantage here. The comparison can be made "fair" by adding a state parameter to the conventional HMM. But which one to choose? So in the end we leave the two systems as they are. Further investigation of this issue will be presented in 9.1.

6.3 Summary of Text-Dependent Verification Results

In summary, there is some evidence from this experiment that an SHMM-based text-dependent speaker verification system can outperform a conventional HMM-based system. As we expected, the better modelling of speech dynamics and duration of the SHMM system helps capture some individual characteristics and hence improve the speaker verification performance.

However, particularly in the case of false rejection errors, the resolution of this test is not sufficiently fine to draw clear conclusions. Therefore it was decided that a more difficult speaker-verification task should be attempted, namely text-independent speaker verification on the Switchboard corpus.

The next chapter presents the methods and experiments used for text-independent speaker verification.

CHAPTER 7

Text-Independent Speaker Verification

As in text-dependent speaker verification, to test the hypothesis that a sequence of acoustic feature vectors $Y = [y_1, y_2, ..., y_T]$ was spoken by a talker S, the likelihood ratio

$$L(S) = \frac{p(Y\ S)}{p(Y)} \tag{7.1}$$

is computed and compared with a pre-determined threshold. A GMM system in used in this case. The probability $p(Y)$ is computed using a background model, which is a GMM trained on acoustic feature vectors corresponding to speech produced by a large population of talkers. The value of $p(Y\ S)$ is computed using a speaker model for speaker S, which is trained on acoustic feature vectors derived from speech produced by S (or, more normally, adapted from the UBM). The quantity $L(S)$ in equation (7.1) is an approximation to the posterior probability of S given the data Y, where the prior probability $P(S)$ of speaker S is ignored. The score $L(S)$ is often normalized using T-norm to allow the same threshold to be used for all talkers

(Auckenthaler et al. 2000).

7.1 A 'segmental GMM'

In order to compare conventional methods with a SHMM-based method for text-independent speaker verification, it is natural to attempt to construct a segmental HMM version of a conventional GMM based speaker recognition system.

In a GMM-based system:

- A speech signal is treated as a sequence $Y = [y_1, y_2, ..., y_T]$ of independent acoustic feature vectors,
- $p(Y)$ is computed as a product of probabilities $p(y_t)$, $p(Y) = \prod_{t=1}^{T} p(y_t)$, and
- Each $p(y_t)$ is evaluated using a weighted sum of multivariate Gaussian PDFs defined on the acoustic feature space.

By analogy, in our 'segmental GMM':

- Y will be treated as a sequence of K independent segments, $Y = \left[Y_1^{t_1}, Y_{t_1+1}^{t_2}, ..., Y_{t_{K-1}+1}^{N}\right]$ (where K depends on Y),
- $p(Y)$ is computed as a product of probabilities $p(Y_{t_{k-1}+1}^{t_k})$, $p(Y) = \prod_{k=1}^{K} p(Y_{t_{k-1}+1}^{t_k})$, where $t_0 = 0$ and $t_K = N$, and,
- Each $p(Y_{t_{k-1}+1}^{t_k})$ is evaluated using a trajectory-based segment model

Since the number of segment boundary points K and the values of the boundary points $t_1, t_2, ..., t_K$ are not known in advance, they must be calculated during the speaker-verification process using the segmental Viterbi decoder. By employing a segmental variant of the forward-backward algorithm for conventional HMMs, it would be possible to calculate $p(Y)$ by summing over all possible values of K and segmentations $t_1, t_2, ..., t_K$, and for an individual segment $[t_{k-1}+1, t_k]$ to calculate $p(Y_{t_{k-1}+1}^{t_k})$ by summing over all segment models. However, in the present study this was discounted on computational grounds, and also for the practical reason that it would necessitate substantial development of additional software within the SEGVit toolkit. Instead we use the segmental Viterbi decoder to find the optimal value of K and segmentation $t_1, t_2, ..., t_K$, and for each segment $[t_{k-1}+1, t_k]$ we define $p(Y_{t_{k-1}+1}^{t_k}) = max_\sigma p(Y_{t_{k-1}+1}^{t_k}\ \sigma)$, where σ ranges over all possible segment model states.

In terms of a conventional GMM, this is analogous to computing the acoustic vector probability $p(y_t)$ by

$$p(y_t) = max_{m=1,...,M} p_m(y_t) \tag{7.2}$$

rather than by

$$p(y_t) = \sum_{m=1}^{M} p_m(y_t) \tag{7.3}$$

That is by choosing the best Gaussian component in the GMM instead of summing over all components. Auckenthaler's work also describes methods of choosing a subset of all components or the best Gaussian component to reduce computation (Auckenthaler 2001). For consistency, and in order to

focus on the 'frame-based' versus 'segment-based' comparison which is the subject of this research, we use equation (7.2) rather than (7.3) in all of our 'baseline' GMM experiments. Once this decision has been made, it will be seen that a conventional GMM is equivalent to a 'segmental GMM' in which the maximum segment duration τ_{max} is set to 1.

7.2 Construction of the 'segmental GMM'

Intuitively, the most natural approach to the problem of applying SHMMs to text-independent speaker verification is to replace the conventional GMM with a single 'segmental GMM'. The segmental GMM consists of M states, each associated with the type of variable-duration linear trajectory segment model described in Section 5 , specified by mean, slope and variance vectors in the acoustic space and a duration probability distribution. These states are configured in parallel, with a single initial, non-emitting, 'null' state and a single non-emitting final 'null' state (Figure 7.1). The segmental states are analogous to the mixture components in a conventional GMM system, and the transition probabilities from the initial null state to the emitting states correspond to the GMM component weights. While a conventional GMM system analyses each acoustic feature vector in a speech signal separately, a segmental system attempts to model the speech signal as a sequence of variable length acoustic segments.

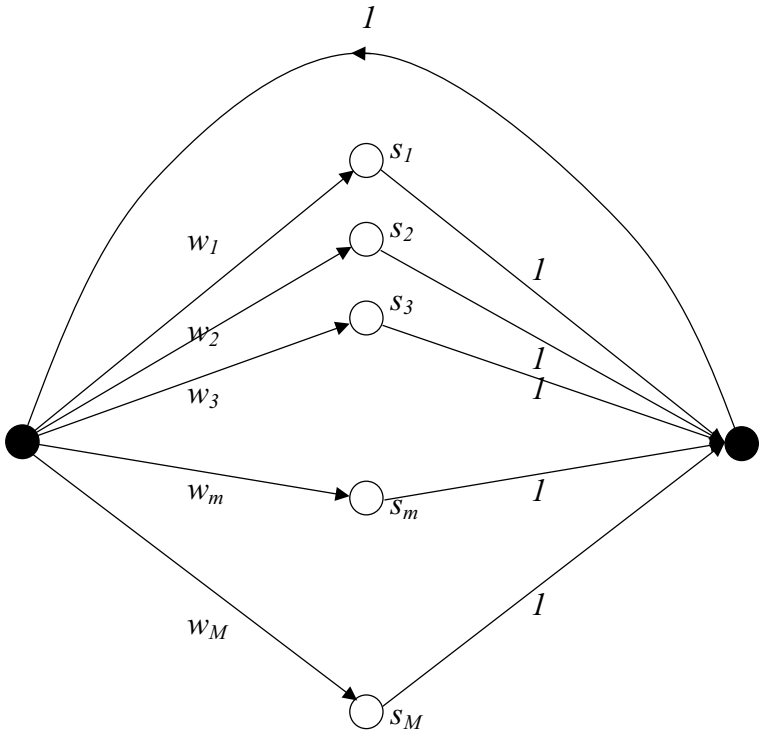


Figure 7.1: SHMM structure for text-independent speaker verification.

7.2.1 Probability Calculations

Given a sequence $Y = [y_1, y_2, ..., y_T]$ which is claimed to correspond to an utterance spoken by speaker S, we compute the likelihood ratio:

$$L(S) = \frac{p(Y\ S)}{p(Y)} \tag{7.4}$$

where the speaker-dependent probability $p(Y\ S)$ is given by:

$$p(Y\ S) = max_K max_{t_1,t_2,...,t_K} max_{\sigma^S_{i(1)},...,\sigma^S_{i(K)}} \prod_{k=1}^{K} ({w^S_{i(k)}}^{\lambda_1} p(Y^{t_k}_{t_{k-1}+1}\ \sigma^S_{i(k)})\lambda_2) \tag{7.5}$$

In other words, for the speaker-dependent probability $p(Y\ S)$ the maximum is taken over all possible numbers of segments K, all possible segmentations $t_1, t_2, ..., t_K$ of length K, and all possible sequences of length K $\sigma^S_{i(1)}, ..., \sigma^S_{i(K)}$ of states from the speaker-dependent model for speaker S. λ_1 is the Language Model Scale Factor (LMSF) and λ_2 is the Token Insertion Penalty (TIP). The LMSF and TIP parameters are commonly used in conventional HMM systems (Woodland et al. 1995).

Similarly the UBM probability $p(Y)$ is given by:

$$p(Y) = max_J max_{t_1,t_2,...,t_J} max_{\sigma^U_{i(1)}BM,...,\sigma^U_{i(J)}BM} \prod_{j=1}^{J} ({w^U_{i(j)}BM}^{\lambda_1} p(Y^{t_j}_{t_{j-1}+1}\ \sigma^U_{i(j)}BM)\lambda_2) \tag{7.6}$$

For the background probability $p(Y)$ the maximum is taken over all possible numbers of segments J, all possible segmentations $t_1, t_2, ..., t_J$ of length J, and all possible sequences of length J $\sigma^B_{i(1)}, ..., \sigma^B_{i(J)}$ of states from the background model. We use different letters (K and J) for the segment se-

quence lengths in equations (7.5) and (7.6) to emphasize that, in general, both the number of segments and the segment index will be different for the speaker-dependent and background-model probability calculations.

7.2.2 The Language Model Scale Factor λ_1 and Token Insertion Penalty λ_2

The effect of the LMSF λ_1 is to control the influence of the individual 'mixture weights' $w^S_{i(k)}$ and $w^B_{i(j)}$ in equation (7.6). A large value of λ_1 will 'sharpen' the distribution $\left[w^B_1, w^B_2, ..., w^B_M\right]$ and increase the influence of the weights. Conversely, if $\lambda_1 = 0$ then the weights will have no effect at all. The TIP λ_2 is a multiplicative penalty which is incurred each time a new segment is hypothesized. An explanation of a sequence Y which involves K segments will incur a penalty of ${\lambda_2}^K$. Thus setting $\lambda_2 = 1$ will have no effect, but setting $\lambda_2 > 1$ will favor longer sequences and setting $\lambda_2 < 1$ will favor shorter sequences.

In the 'SEGVit' SHMM toolkit, all probability calculations are done in the negative logarithmic domain (where maximizing a probability is translated into minimizing a cost), and parameters such as the LMSF and TIP are specified in the configuration file as values in that domain. In the negative logarithmic domain λ_1 becomes a multiplicative factor and λ_2 becomes an additive penalty. With respect to this domain, setting $\lambda_1 = 1$ and $\lambda_2 = 0$ will have no effect. So the default values in the 'SEGVit' SHMM toolkit for the LMSF λ_1 is 1 and for the TIP λ_2 is 0. Setting the LMSF parameter $\lambda_1 > 1$ increases the effect of the weights in choosing which segment model to

use. Setting $\lambda_2 > 0$ will favor shorter segment sequences (and hence longer individual segments) and setting $\lambda_2 < 0$ will favor longer sequences (and hence shorter individual segments). Thus the TIP parameter λ_2 provides an external mechanism for influencing segment lengths.

7.3 Comparison of computational loads for GMMs and SGMMs

7.3.1 GMM computational load

Suppose that we have an M component GMM and an utterance $Y = y_1, y_2, ..., y_t, ..., y_T$, each y_t is of dimension d. First let us just consider one vector y_t. For each t, we need to do M d-dimensional log Gaussian probability calculations (LGPCs), plus $M - 1$ pairwise comparisons to find the maximum. So we need $T * M$ d-dimensional LGPCs plus $T * (M - 1)$ pairwise comparisons plus $(T - 1)$ additions.

7.3.2 SGMM computational load

Similarly, suppose that we have an M component segmental GMM and the same utterance $Y = y_1, y_2, ..., y_t, ..., y_T$. The maximum duration parameter D_{max} is set to 15 for the segmental GMM. The calculation requires segmental Viterbi decoding:

$$\hat{\alpha}_t(i) = max_D max_j \hat{\alpha}_{t-D}(j) a_{ij} b_i(y_{t-D+1}, y_{t-D+2}, ..., y_t) \tag{7.7}$$

in which $1 \leq D \leq D_{max}$.

First let us fix D and j, and remember this is all done in the log domain. $b_i(y_{t-D+1}, y_{t-D+2}, ..., y_t)$ requires D LGPCs. Because the trajectory is re-calculated for every D there is no easy way to re-use the LGPCs, so the total number of LGPCs is

$$1 + 2 + 3 + ... + D_{max} = \frac{(D_{max} + 1)D_{max}}{2} = \frac{16 * 15}{2} = 120 \tag{7.8}$$

In addition, for each segment length the trajectory means have to be calculated. The number of pairwise comparisons is $(M-1)D_{max}$ (finding the maximum over j and the maximum over d). All of these has to be done for $t = 1, 2, ..., T$.

7.3.3 Comparison

So, the basic comparison is between, for each t, the computation is M LGPCs + $(M-1)$ pairwise comparisons for GMM, and $M\frac{(D_{max}+1)D_{max}}{2} = 120M$ LPGCs plus $M(M-1)D_{max}$ pairwise comparisons for segmental GMM, as shown in the Table 7.1.

Therefore, for example, if D_{max} is increased from 15 to 16, the number of LGPCs needed for the segmental GMMs increases from $T * M * 105$ to $T * M * 120$. That is $15 * T * M$ more LGPCs. If the number of segmental GMM components M is set to 300, there are $4500 * T$ more LGPCs for each t. As we can see, the computational load for the segmental GMMs is huge compared to the computational load for the conventional GMMs.

Table 7.1: Computational loads comparison

	GMM	SGMM
number of LGPCs	$T * M$	$T * M * \frac{(D_{max}-1)D_{max}}{2}$
number of pairwise comparisons	$T * (M-1)$	$T * M * (M-1) * D_{max}$
other		trajectory calculations (working out sequence of means in trajectory) $T * M * D_{max}$

We tried various methods to speed up the experiment turn-around time for the segmental GMM system. They will be mentioned in 7.4.4.

7.4 Experiment methods

7.4.1 Switchboard data sets used

The 2002 (Linguistic Data Consortium 2002) and 2003 (Linguistic Data Consortium 2003) NIST SRE subsets of Switchboard were obtained through NIST and LDC to enable us to evaluate the segmental GMM for speaker detection on the NIST 2003 SRE test. The experiments use:

- The one-speaker training material from the 2002 NIST SRE to train the UBM,
- The one-speaker training data from the 2003 NIST SRE to train the SDMs, and
- A subset of approximately 50% of the one-speaker test data from the 2003 NIST SRE as test data. Only 50% of the data was chosen to reduce the computational load and the whole experiment running time.

An analysis of the systems used in the 2003 NIST SRE and the results obtained suggests that a suitable parameterization of the speech signal would comprise mel frequency cepstral coefficients 1 to 18, plus energy, plus the corresponding Δ parameters (National Institute of Standards and Technology 2003). However, in the present system only the static parameters were used. This was partly to reduce the computational load, and partly because it was hoped that explicit modelling of speech dynamics would remove the need for the Δ parameters, as discussed earlier in section 6.1.1. The data was parameterized as 18 mel frequency cepstral coefficients (MFCCs) plus an energy measure (C0) using the HTK 'HCopy' tool[1].

7.4.2 The model training

Experience from conventional GMM systems on Switchboard suggests that an appropriate number of segmental UBM components is at least 1024 (National Institute of Standards and Technology 2003; Reynolds et al. 2000). However, the time taken to train and evaluate such a model would preclude an extensive investigation of the effect of different SHMM variants and parameters on speaker recognition performance. Auckenthaler's work (Auckenthaler 2001) compares the performances of three systems, with model sizes of 64, 256 or 1024, respectively. His experiments used scoring of the speaker model with either the best, the best three or the best five components for

[1]At first the MFCC-based parameterization which uses an explicit measure of energy was chosen (MFC_E), however it was found that with this parameterization HCopy gives incorrect results — abnormal huge negative numbers — for some of the energy measure parameters of Switchboard data. This is because there are silences in Switchboard data which produces all zero mfcc parameters. The log energy of these silence frames are invalid values. This problem does not occur if the zeroth MFCC coefficient (MFC_0) is used instead

each different model sizes. The results show that for the model size of 1024 components the system degraded when only the best scoring component is used. While for model sizes of 64 and 256 components the models perform best when only the best scoring component is used. As we were doing a comparative experiment, smaller model size provides us a faster running cycle which leaves freedom to investigate different system settings. Hence, for the current experiments the number of components in both GMMs and SHMMs was set to 300.

Both GMM and segmental GMM systems were constructed using the SEGVit software. The GMMs were constructed with the segment durations fixed to one and trajectory slopes set to zero. The segmental GMMs have non-zero trajectory slopes and a non-parametric duration probability function. Model training and determination of the optimal segment sequence use segmental Viterbi decoding.

'Segmental GMM' UBM construction for Switchboard

As part of the previous research on TIMIT phone classification (Russell and Jackson 2005), a software has been developed to produce sets of context-sensitive triphone SHMMs of varying sizes (using the monophone and biphone 'back off' approach described earlier). Using this software we have developed TIMIT-based model sets with between 104 and 5,989 models (or, equivalently, between 312 and 17,967 states). By combining the states of a suitable family of models into a single, integrated SHMM of the type depicted in Figure 7.1 we hoped that we could obtain a suitable initial model to 'seed' Viterbi re-estimation of our segmental UBM (Figure 7.2). Estimation

of the target speaker models could then proceed as previously described. For this pilot experiment we used the 104 TIMIT model set to form a 312-state segmental GMM. The maximum segment duration τ_{max} was set to be equal to five.

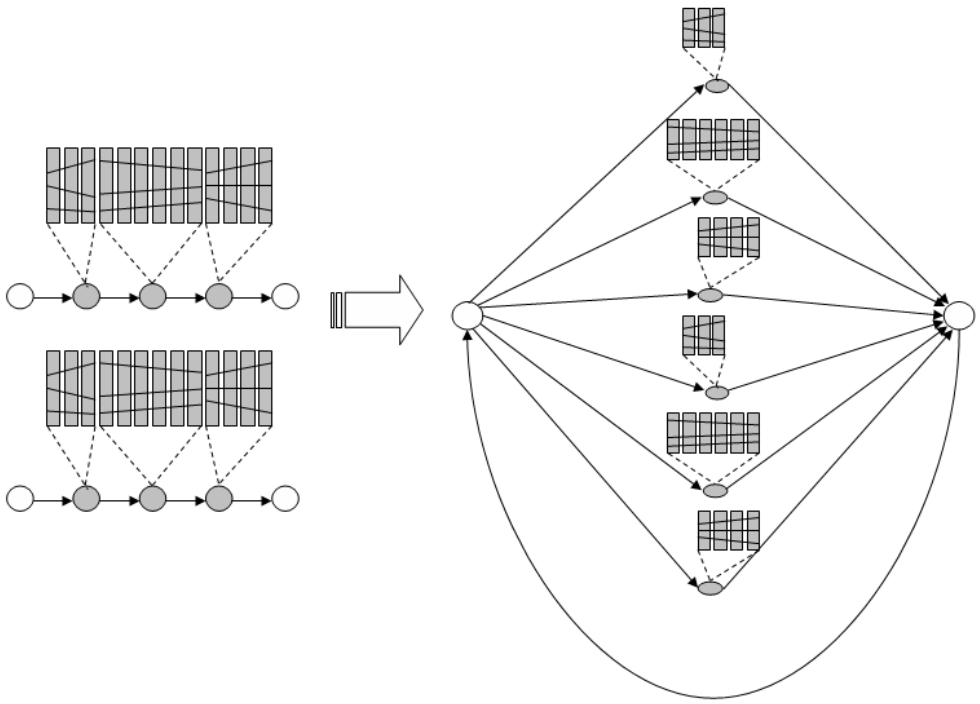


Figure 7.2: Segmental GMM construction from TIMIT trained HMMs.

Unfortunately this did not prove to be the case. The dissimilarity between the TIMIT-based models and the Switchboard data was such that nearly 80% of the 312 SHMM states were not used during re-estimation. After two iterations of the reestimation, only 20% of the SHMM states had non-zero 'occupancy' and could therefore be reestimated. Thus the effective number of states was significantly reduced. We concluded that it it not pos-

sible to use segmental states estimated on TIMIT as initial models for work on Switchboard because of the significant differences between TIMIT and Switchboard data.

As an alternative we used k-means clustering (MacQueen 1967) applied to a randomly chosen subset of the Switchboard 2002 data to estimate the 300 segment means. The initial segment trajectory slope values were set to zero and the state duration distributions were set to be uniform. Using this method to initialize the UBM, all of the UBM states were re-estimated during segmental-Viterbi based training afterwards. Based on these k-means clustering estimated initial models, five iterations of segmental Viterbi training algorithm were applied before the process converged, with the segment duration distributions only being re-estimated during the final iteration.

These initial segment models were used to construct an initial 'segmental GMM' background model, which was optimized using the Viterbi-based SHMM re-estimation functions in the SEGVit toolkit and the NIST 2002 SRE one-speaker training set. There is speech from 330 speakers (191 females and 139 males) in the 2002 SRE one-speaker training set. Each speaker has roughly 2 minutes of speech. During training the segment trajectory means and variances were re-estimated first, using four iterations of Viterbi training algorithm. Then the segment trajectory means, slopes and variances were re-estimated for a further five iterations. The duration probabilities were only re-estimated in the final, fifth, iteration.

Different maximum segment lengths corresponding to $\tau_{max} = 1, 5$ and 10 were chosen to make three sets of models, which we refer to as $SW1$, $SW5$, and $SW10$. These models were built to test the effect of maximum segment

duration on speaker-verification performance. For all model sets except $SW1$, the segment trajectory means and slopes, variances and the segment duration distributions were estimated. In the case of $SW1$, only the segment trajectory means and variances were re-estimated, the trajectory slopes were set to zero and the duration length can only be one frame. $SW1$ was treated as the counterpart of the traditional GMM system and used as our baseline system. The advantage of doing this for a comparative experiment is that we can be certain that we are controlling the differences between the systems.

Training procedure for the speaker-dependent 'Segmental GMMs'

Each trained UBM was then used to seed a speaker-dependent 'segmental GMM' for each of the test speakers in the 2003 Switchboard test set. Data from the 2003 Switchboard training set was used to re-estimate these models. The data include 149 male files and 207 female files, each file containing about 2 minutes training data. The SDMs of 3 different UBM sets $SW1$, $SW5$ and $SW10$ were trained separately with duration $\tau_{max} = 1, 5$ and 10 frames. For the UBM set $SW5$ ($\tau_{max} = 5$), another three different sets of SDMs were produced:

- In scheme 1, $SW5\_1$, the segment trajectory mean vectors were re-estimated but the slope vectors were set to zero in both the UBM and SDMs.

- In scheme 2, $SW5\_2$, only the segment trajectory mean values were re-estimated. The segment trajectory slopes in the SD models are therefore the same as those of the corresponding segment models in

the UBM.

- In scheme 3, $SW5\_3$, the segment trajectory slopes were also re-estimated, along with the segment trajectory means in the SD models.

As the trajectory slopes should contain some dynamic information from the speaker, we expected scheme 3 to outperform scheme 2. Both schemes should outperform scheme 1, which does not include any dynamic information.

For the speaker-dependent models the segment duration models and variance parameters were not re-estimated because of the limited amount of training data which is available for each speaker. The trajectory means were re-estimated in all cases. No speaker adaptation method, such as MAP or MLLR, was used. MAP or MLLR is not implemented in SEGVit.

7.4.3 Factors influencing the performance of a 'segmental GMM'

In summary, there are some key parameters of the 'segmental GMM', whose effect on verification performance we want to measure.

The maximum segment duration

The parameter τ_{max} specifies the maximum allowable segment duration. If $\tau_{max} = 1$ then states are associated with individual feature vectors, and our 'segmental GMM' reduces to a type of conventional GMM. As τ_{max} increases, the model becomes 'more segmental' but the computational load increases.

In our experiments on Switchboard, values of 1, 5 and 10 were chosen for τ_{max}.

The trajectory slope

This could be set to zero, estimated for the UBM from training data and then maintained at this value for each speaker-dependent model, or reestimated for each speaker model. The significance of the trajectory slope parameters is likely to depend on the τ_{max} parameter: with slope being more significant for larger values of τ_{max}.

The segment duration model

Again this could be trained from data for the UBM and either passed unchanged to each speaker-dependent model or reestimated for each speaker-dependent model. Since duration is a segment-level, rather than frame-level, parameter, very few training examples of segment duration are likely to be contained in a typical speaker-dependent adaptation or training set. Therefore accurate estimation of a speaker-dependent duration model is likely to be an issue.

The language model control parameters λ_1 and λ_2

As explained previously, the SEGVit system includes two parameters, LMSF (λ_1) and TIP (λ_2) which can be used to influence the average segment duration. If λ_1 and λ_2 take their default values of 1 and 0, respectively (remember that these parameters operate in the negative log probability domain), then they have no effect on the Viterbi decoder. However, by adjusting these two

control parameters away from their default values during training or testing, it is possible to influence the durational structures of the segments in the UBMs and SDMs.

Figure 7.3 shows the effect of varying the TIP parameter, λ_2 on segment duration statistics. In these experiments λ_1 was set to 1 while λ_2 was varied between -10 and 100. It is important to note that these statistics are obtained from the test data. The UBM and SDMs were trained with $\tau_{max} = 10$, $\lambda_1 = 1$ and $\lambda_2 = 0$.

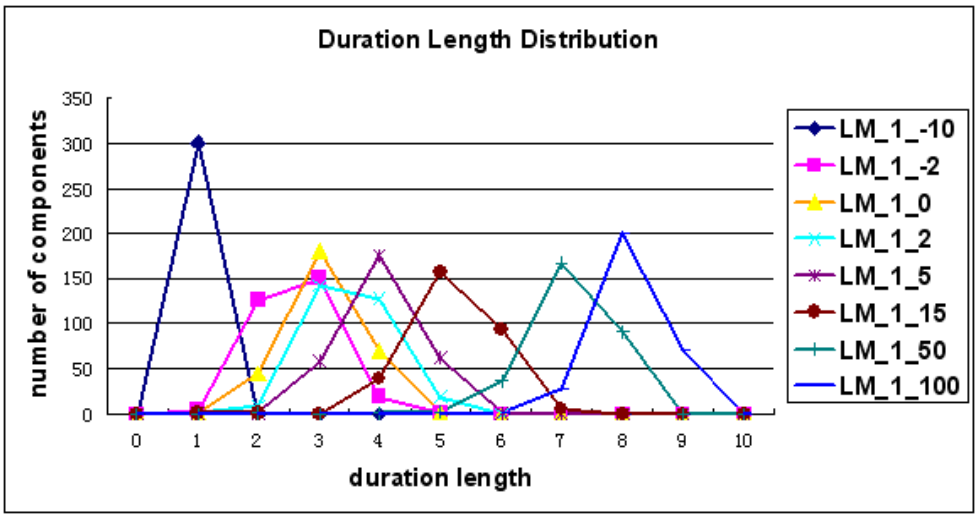


Figure 7.3: Duration length distributions for different values of the Token Insertion Penalty λ_2. LM_x_y refers to the case where $\lambda_1 = x$ and $\lambda_2 = y$. LM_1_0 is the default..

Figure 7.3 shows that the average segment duration for the 'default' case where $\lambda_1 = 1$ and $\lambda_2 = 0$ is 30ms, with a minimum duration of 5ms and a maximum duration of 70ms. By increasing λ_2 to 100 the most probable duration is increased to 80ms. For such large values of λ_2 it is likely that there is a conflict between the effect of λ_2, which is to increase the expected

segment duration, and the hard upper-bound on segment duration imposed by τ_{max}. Setting $\lambda_2 = -2$ shifts the duration distribution slightly to the left (towards shorter durations), while setting $\lambda_2 = -10$ causes all segments to have minimum duration, which is 10ms (or one acoustic vector).

The number of segments

Most state-of-the-art GMM based TI-SV systems use 2048 or at least 1024 components. From Auckenthaler's work on Switchboard (Auckenthaler 2001), actually no significant degradation of performance is shown for model sizes of 64 and 256 components, compare to a model size of 1024. We did not investigate exactly how different numbers of segments in the system would affect the SV performance of our segmental GMMs. However, to save the computational load for our segmental GMM system, which allows different system settings being tested, we only use 300 segments in our segmental GMM based system.

7.4.4 Speeding Up Experiment Turn-around Time

As mentioned in 7.3, because of the need to run segmental Viterbi decoding and to compute segment-level probabilities, the computational load associated with our 'segmental GMM' is significantly greater than that associated with a conventional GMM. Experiments that can be done within days using a conventional GMM can run for a month using the segmental GMM, depending on the segmental system settings.

In order to reduce this computational cost, speaker-verification experi-

ments were conducted using just half of the male test speakers (671 speakers) and half of the female test speakers (1042 speakers) from the NIST 2003 single-speaker evaluation set. This reduces the computation in testing by 50%.

As mentioned earlier, the number of segmental states in the model was also kept low at 300. However, the computational load was still prohibitive. We applied the following techniques to improve the experimental turn-around time.

Parallelization of the SEGVit toolkit

The 'SEGVit' software toolkit has been modified so that model training can be conducted in parallel on a 'grid' of computers. However the computation time is still prohibitively long for a large detection task. For example, we estimate that an evaluation of our reduced system, with $\tau_{max} = 15$ will take between 20 and 25 days on our 6-node cluster, while the same experiments with a conventional GMM system using HTK only takes less than a week to finish.

Beam pruning and duration pruning

Techniques which work for standard HMMs, such as Beam Pruning (Russell 2005) have been extended to the 'SEGVit' toolkit during the period of this project. Beam pruning uses a beam threshold to prune any preceding state if the margin between the forward probability of the preceding state and the forward probability of the present state is smaller than the threshold. In our experiments beam pruning was shown to be much less effective for

speaker detection than for speech recognition. This is because at present there is effectively no syntax to constrain possible segment sequences. In other words, because each segment in the 'segmental GMM' can be preceded by every other segment, pruning out paths in the past does not alter the number of segments which have to be evaluated in the present.

Russell developed a technique which we refer to as 'Duration Pruning' (Russell 2005) whereby a segment probability is not calculated if the probability of its duration is below a pre-determined threshold. Again, this technique works well for phone recognition experiments on TIMIT but appears to be less useful for speaker verification experiments on Switchboard.

Auckenthaler's method for reducing computational load

In a further attempt to speed up our experiments, I investigated a technique introduced by Auckenthaler (Auckenthaler 2001). The method exploits the link between the UBM and each of the SDMs. Since each SDM is seeded by the UBM, it is argued that there is a strong connection between the m^{th} component of the UBM and the corresponding m^{th} component of the SDM. Thus, once the optimal sequence of components has been computed for the UBM, Auckenthaler used exactly the same sequence for each of the SDMs.

I developed new software within the SEGVit toolkit to implement my analogy to Auckenthaler's method. During recognition the sequence of best components was computed for the UBM. The same sequence was used for each of the 1713 SDMs. As mentioned in 7.3, calculating each possible duration for a segment during probability calculation takes long computation time. By using Auckenthaler's method, the duration probability was only

calculated once, for the UBM. The duration probability for each of the 1713 SDMs did not need recalculation as the UBM sequence was used for each of the SDMs. The new method effectively reduced the whole processing time for training and testing for a segmental GMM system with $\tau_{max} = 15$ from around one month to one week using the SEGVit software toolkit on our 6-node computer cluster, with almost no loss in system performance, compared to the previous experiment result without using Auckenthaler's method.

7.4.5 Speaker Verification Experiments

As mentioned previously, in order to reduce this computational cost and to improve experimental turn-around time, speaker-detection experiments were conducted using just half of the male test speakers (671 speakers) and half of the female test speakers (1042 speakers) from the NIST 2003 single-speaker evaluation set.

As specified in the NIST 2003 evaluation documentation, for each test file, 11 different verification tests were performed. This in turn involved 12 probability calculations - one for the background model and 11 for the speaker-dependent models specified in the NIST test set.

The following experiments were conducted:

- **Experiment 1**: This experiment investigated the effects on performance of setting the trajectory slope values to zero in both the UBM and SDMs ($SW5\_1$), reestimating the trajectory slope vector for the UBM but not for the SDMs (so that the SDM trajectory slope vectors are equal to the corresponding UBM slope vectors, $SW5\_2$), and rees-

timating the slope vector for both the SDMs and the UBM ($SW5\_3$). In this experiment $\tau_{max} = 5$.

- **Experiment 2**. The performances of the systems with maximum duration τ_{max} set to 1 ($SW1$), 5 ($SW5$) and 10 ($SW10$) were compared. In these experiments all of the UBM trajectory parameters were reestimated and used to seed the corresponding SDM parameters, and all of the SDM parameters were then reestimated (except in the case of $SW1$, where the slope vectors are all zero - this is the baseline system)

7.5 Results of text-independent speaker verification experiments on Switchboard

7.5.1 Effect of the trajectory slope vector

The results for the first experiment (**experiment 1**), with model sets $SW5\_1$ $SW5\_2$ and $SW5\_3$ are shown as DET curves in Figure 7.4a. Conditions 1, 2 and 3 correspond to trajectory slopes set to zero in the UBM and SDMs; UBM trajectory slopes learnt but not reestimated in the SDMs; UBM trajectory slopes learnt and reestimated for the SDMs respectively. In these experiments $\tau_{max} = 5$. The figure shows that the equal error rate for all three systems is approximately 14%. The best performance is obtained using speaker-dependent trajectory slopes (scheme 3 - red dashed line), but the difference between this and the other results (trajectory slopes set to zero (scheme 1 - black dotted line), trajectory slopes re-estimated for the UBM but not re-estimated for the SDMs (scheme 2 - green solid line) is very small

and unlikely to be significant.

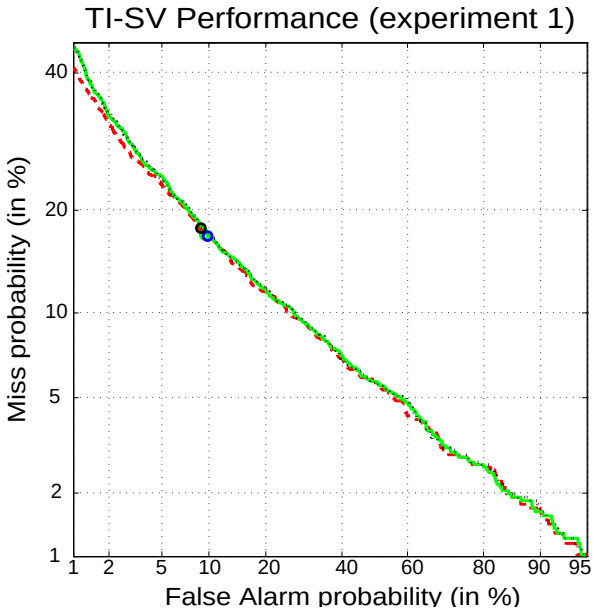
7.5.2 Effect of the maximum segment duration τ_{max}

The results of the second experiment (**experiment 2**), for systems with different maximum durations, namely $SW1$ ($\tau_{max} = 1$), $SW5$ ($\tau_{max} = 5$) and $SW10$ ($\tau_{max} = 10$) are shown in Figure 7.4b. $SW1$ is our approximation to a conventional GMM. The figure shows that the systems with $\tau_{max} = 5$ (scheme 2 - black dotted line) and $\tau_{max} = 10$ (scheme 3 - green solid line) work very slightly better than the system with $\tau_{max} = 1$ (scheme 1 - black dashed line), but still the differences are too small to be significant.

Other experiments have been conducted using different SHMM variants and parameters, for example, further adjusting the maximum duration, and altering the statistics of segment durations. However, all the performances are very close and the equal error rates are all approximately 14%.

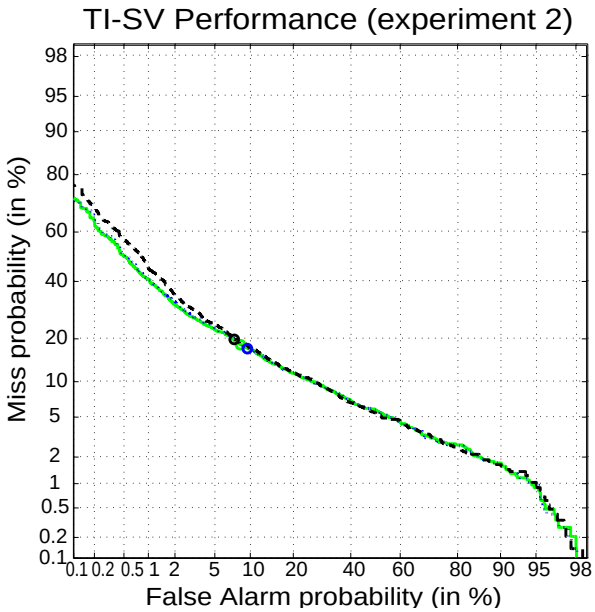
7.5.3 State-of-the-art TI-SV systems on NIST SRE 2003

It is helpful to have a look at what performances state-of-the-art TI-SV systems achieve for the 2003 NIST Speaker Recognition Evaluation (National Institute of Standards and Technology 2003). In this thesis the same data was used except for that only half of the test segments (1713 of 3428 segments) were used to save the computation. Figure 7.5 shows the results from the NIST 2003 Speaker Recognition Evaluation participants. From the figure we can see that most of the systems managed to gain an Equal Error Rate



a. experiment 1

SW5_1. black dotted line, SW5_2. green solid line, SW5_3. red dashed line



b. experiment 2

SW1. black dashed line, SW5. black dotted line, SW10. green solid line

Figure 7.4: TI-SV Results on Switchboard using GMMs and SHMMs.

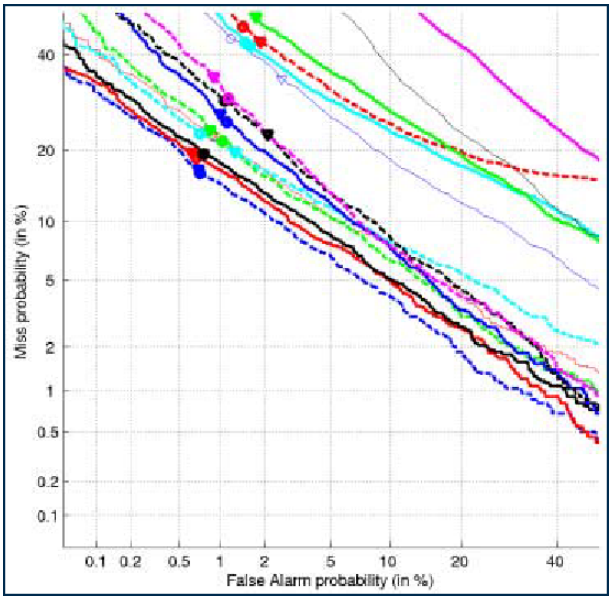


Figure 7.5: NIST 2003 Evaluation Results

(EER) between 5% and 10%. Some systems' performance fall in the EER range between 10% and 40%.

The best speaker verification performance on the 2003 Switchboard data is given by the MIT Lincoln Laboratory, which is just above 5% EER. This performance is gained using a 2048 mixture GMM, feature mapping (Reynolds 2003), RASTA filtering (Hermansky and Morgan 1994), speech activity detection (Appiah, Sasikath, Makrickaite, and Gusaite 2005), Support Vector Machines (Campbell et al. 2006), and Biologically-Inspired Auditory Features (National Institute of Standards and Technology 2003). The Directorate of Defense Research & Development (DDR&D) system also produce an EER a little bit above 5%. Their system has 2048 Gaussian mixtures, feature warping (Pelecanos and Sridharan 2001), voice activation detector, Principal Component Analysis (PCA) and Autoassociative Neural Network

(Shajith et al. 1999).

Comparatively our system with an EER around 14% doesn't compete with the systems with best performances. The NIST figure shows a "peel" of about 9 systems with EERs between 5% and 10% and about 6 other systems with EERs between about 12% and 25%. Our system isn't in with the highly optimized systems in the peel, but is better than some of the others.

Our system has only 300 states, while other systems have 1024 or 2048 GMM components. However, from Auckenthaler's work using the 1999 Switchboard data (National Institute of Standards and Technology 2003) it seems that the speaker verification performance of an 256-component GMM system is not much worse than the SV performance of an 1024-component GMM system, considering other conditions in both systems are exactly the same. The reason that our system does not perform as well as these systems is probably due to the different front end processing and the absence of SVM. Our system does not use speech/noise detector, RASTA filtering, or feature warping. These methods were used by other systems to remove the effect of noise in the Switchboard data. Without using any noise compensation technique, the performance of the system is expected to be much worse than those refined systems. The SVM may also be a major contribution to higher speaker verification performance.

7.5.4 Summary of Text-Independent Speaker Verification Results

These results are certainly not as we expected. We expected that in experiment 1 scheme 1 would give poorer results than schemes 2 and 3, and thereby demonstrate the utility of modelling dynamics by incorporating a non-zero slope parameter. In fact this experiment provides little evidence to support the hypothesis that the use of linear segment models with non-zero trajectory slopes is beneficial for speaker detection. This result contrasts with the previous result for YOHO, where there does appear to be a benefit.

In the second set of experiments we expected that $SW10$, with maximum segment duration set to 10, would outperform $SW5$ ($\tau_{max} = 5$), and that $SW5$ would in turn outperform $SW1$ ($\tau_{max} = 1$). However there is little evidence in the results to support this expectation. It should also be noted that the results of experiments 1 and 2 are consistent. If, as suggested by the results of experiment 1, there is no benefit from using a model based on 'dynamic' trajectories with non-zero slope, then one would not expect to observe any benefit from longer segments, since a long, constant segment can be modelled just as well by a sequence of short, constant segments. The duration probability density functions are clearly different in these two cases. However the duration PDFs are usually reestimated at the last step of training, when the mean and slope of the segment trajectories are already well trained.

We note that all of these results are clearly much worse than the best performance obtained on the full 2003 test set using a conventional GMM

system, which is a little over 5% equal error rate (Martin and Przybocki 2003). The poor performance of our system is likely to be due to different front-end analysis. We did not use any noise compensation technique or the Support Vector Machines in our system. To cut the computational load, during probability calculation, we choose the best Gaussian component in the GMM instead of summing over all componnets, which should also affect the system performance. However, the goal of these initial experiments was not to challenge the state-of-the-art in terms of performance, but to conduct comparative experiments to determine the benefits of using a dynamic, trajectory-based model.

Effects of reducing the computational load

The results obtained by applying the 'segmental GMM' version of Auckenthaler's method, described in section 7.4.4, are shown in the DET curves in Appendix A. Each figure shows two DET curves. The dashed (blue) line is the same in all of the figures and is included as a baseline. It shows the DET curve obtained when separate Viterbi decoding is applied to each of the SDMs (i.e. Auckenthaler's method is not used). For these experiments $\lambda_1 = 1$, $\lambda_2 = 0$ and $\tau_{max} = 10$.

The solid (red) DET curves show the results of applying Auckenthaler's method (i.e. using the optimal state sequence obtained using Viterbi decoding relative to the UBM to calculate the SDM probabilities) together with different values of language model control parameters λ_1 and λ_2 ($\lambda_1 = 1; \lambda_2$ $-10, -2, 0, 2, 5, 15, 50, 100$).

Figure A.1 shows a direct comparison, for $\lambda_1 = 1$ and $\lambda_2 = 0$, of the re-

sults obtained with and without the computational reduction due to Auckenthaler's method. The figure shows that the DET curves are almost identical, with the reduced computation method showing small gains at each extreme of the DET curve but performing slightly worse towards the center of the curve. We conclude that the large reduction in computational load which results from using the optimal UBM state sequence to calculate the SDM probabilities is not compromised by a significant change in speaker detection performance.

Turning now to the effects of varying the Token Insertion Penalty λ_2 (figures A.7 to A.8) we see that there is very little difference between the DET curves for the different values of λ_2, despite the large variation in expected segment duration shown in figure 7.3. In particular, it is certainly not the case that (as one might have expected) performance reaches a maximum for some positive value of λ_2. Indeed, larger values of λ_2 lead to decreases in performance, and the best performance is obtained with $\lambda_2 = -2$. From figure 7.3 this value of λ_2 corresponds to an expected segment duration of between 20ms and 30ms. It seems that shorter segment duration lengths give the best performance, which is quite different from what we expected but consistent with the results for varying τ_{max}.

Effects of applying λ_1 and λ_2

At this point we noted a possible incompatibility in these experiments. The language model control parameter λ_2 was only varied during testing and not during training. Therefore it's effect on segment duration during testing is incompatible with the duration models learnt during training. To make the

effect of the language model control parameters compatible with the model durations, additional experiments were carried out. In these experiments, the language model control parameter λ_2 was the same in model training as in testing ($\lambda_2 \quad 5, 15, 50$).

The results of these experiments are shown in Appendix B. The DET curves for the systems which use the optimal UBM state sequences when calculating SDM probabilities are shown with a solid green line (this is the 'Auckenthaler method'). The DET curves for systems which apply Viterbi decoding separately to the UBM and SDMs are shown with a dashed blue line ($\lambda_1 = 1$; $\lambda_2 = 0$). The DET curve for a conventional GMM system is shown with a solid, black line.

The results are similar to those in Appendix A. These support the hypothesis that the results in Appendix A are not affected significantly by use of different values of λ_2 in training and testing. As in Appendix A, the DET curves in Appendix B show a trend whereby performance decreases as λ_2 (and hence the average segment durations) increases. The figures confirm, again, that Auckenthaler's method has little effect on performance.

7.6 Summary

This chapter has described the construction of a segmental GMM system and the main results on Switchboard data.

The segmental states in a segmental GMM system are analogous to the mixture components in a conventional GMM system. Each segmental state has a trajectory which is defined by a midpoint and a slope vector, and a

duration probability function. While a conventional GMM system analyses each acoustic feature vector in a speech signal separately, a segmental system attempts to model the speech signal as a sequence of variable length acoustic segments. The model training and testing of the segmental GMM system are through a segmental Viterbi decoder.

The background models were seeded using a k-means clustering technique and further trained on the 2002 NIST SRE one speaker material. The speaker dependent models were then trained using UBMs on the 2003 NIST SRE one speaker material. Factors influencing the performance of a segmental GMM include the maximum segment duration τ_{max}, the trajectory slopes, the segment duration model and two language control parameters. These factors are in fact all related to each other. For example, by choosing different values for the LMSF λ_1 and TIPλ_2, the viterbi decoder can be biased towards longer or shorter state sequences and shorter or longer segments.

Some techniques were used to reduce computational time. Auckenthaler's method can effectively cut the testing time. This method recognizes a strong link between a component of the UBM and the corresponding component of the SDMs. Thus the optimal sequence of components computed for the UBM can also be used for each of the SDMs. This method has proved in our experiments to be very effective in reducing computational load without losing significant system performance.

We performed two main sets of experiments, one of which was to test the effect of different segment slopes, the other was to test the effect of different segment lengths. Both results are not as we expected. The first experiment provides little evidence to support the hypothesis that the use of

linear segment models with non-zero trajectory slopes is beneficial for speaker verification. Consistent with experiment 1, experiment 2 doesn't demonstrate any benefit from using potentially longer segments. These results contrast with the YOHO results, which show obvious benefits from using segmental HMMs.

The fact that inclusion of dynamic segments, corresponding to trajectories with non-zero slope, consistently fails to improve speaker detection accuracy on Switchboard, suggests that the segmental GMMs do not contain any important dynamic information which helps to differentiate between speakers in this corpus. But the segmental HMMs trained on a non-conversational corpus like YOHO manage do contain useful information. To find out why, we believe that it is important to conduct further work to determine the exact contribution of dynamic regions of a speech signal to speaker-detection accuracy. The following chapter presents the results of applying these analysis to TD-SV experiments on the YOHO corpus.

CHAPTER 8

Analysis of Text-Independent Speaker Verification system

The results of our speaker verification experiments on Switchboard are not as expected. We have been unable to demonstrate any benefit from the use of 'dynamic' segments based on linear trajectories with non-zero slope. Hence we have also not been able to demonstrate any benefit from the use of longer segments. These results are at odds with our earlier speaker verification results on YOHO, described in section 6.2, and with the phone recognition results presented in (Russell and Jackson 2005).

The discrepancy between the performance of SHMMs for text-dependent detection on YOHO and their performance for text-independent detection on Switchboard is puzzling. There are at least two possible explanations:

- The experiments on YOHO are text-dependent and use the YOHO word-level labeling. This labeling enabled us to use phone-level models in speaker verification. By contrast, no labels were used in the case of

Switchboard and the models were unsupervised 'machine learnt' segment models with no explicit phonetic interpretation. It could be that some sort of explicit labeling is needed to guide the segmental model building process if dynamic regions are to be exploited. Hence the supervised training might have steered the segmental HMMs to model the dynamic regions. In the unsupervised maximum likelihood training, however, the maximum likelihood training criterion seems to bias the system towards stationary states. As mentioned in 5.2, the conventional HMM system uses a mean and a variance to define its MFCC distribution and its delta distribution. To assume both MFCCs and deltas are stationary is not appropriate. Only in the case that the data is stationary can the assumptions of the HMM system be fulfilled. To maximize the probability given these assumptions, the maximum-likelihood training would focus on stationary regions as these regions would produce bigger probability.

- An alternative explanation is that the discrepancy is due to the different styles of speech in the YOHO and Switchboard corpora. While YOHO contains recordings of read speech, Switchboard comprises recordings of conversational speech over various telephone channels. The Switchboard speech is also very noisy. The poorer quality of the Switchboard speech might have caused difficulty for the data-driven segment model learning process, or, alternatively, cues which the segment models were able to use in the YOHO corpus may be absent in Switchboard.

Trying to understand this result, we conducted a set of experiments to

investigate whether the trained segmental GMMs successfully contain speech dynamics and if so, whether this information can contribute to speaker verification performance. Several different parameter sets were used to train the UBMs and the trained UBMs were analysed.

8.1 Visualisation of the segmental GMMs

First, to see what our segmental models look like, we have written a MatLab program to visualize the individual segment models in the segmental GMMs. The results are illustrated in Appendix C.

For each segment, we computed linear trajectories for all 19 MFC coefficients. The length of a segment is its average length, based on its duration distribution. This results in a sequence of 19 dimensional MFCC vectors. We then applied an inverse Discrete Cosine Transform to each of these vectors to obtain a mel frequency spectrum, whose frequency axis was then warped to obtain a linear frequency spectrum. The resulting sequence of linear spectral vectors is displayed as a gray-scale spectrogram to give one of the figures in appendix C.

Visual inspection of these 'spectrograms' suggests that they are all valid speech segments, and that they correspond to different components of a plausible segmental model of speech. For example, the second segment in the third row on the first page of appendix C is clearly vowel like, while the first segment on the fifth row is more fricative-like. The figures show a mixture of stationary and non-stationary segments. However, most of the segments represent the stationary regions. Even in the non-stationary regions the slopes

are very close to zero.

To inspect the issue that whether it is the unsupervised maximum likelihood training or the poor quality of the Switchboard data (or both) that compromises the data-driven segment model learning process, more analyses are to be done. Next section presents the investigations on this issue.

8.2 Segment slopes of UBM trained on Switchboard

Given the TI-SV results in Figure 7.4 and the visualization of the segments, an obvious question is whether the SHMM is actually capturing dynamic information at all. An initial analysis of the values of the slopes in this system suggests that it is not; the majority of slopes are close to zero. One possibility for this is that the unsupervised maximum likelihood GMM training algorithm gives priority to modelling stationary regions. Intuitively, if the number of MFCCs is increased, the independent treatment of the MFCC components will cause more and more states to be used to model stationary regions. To investigate this, we focused on the dynamic behavior of individual, or reduced sets of MFCCs. In these experiments the maximum duration τ_{max} was set to 5 (50ms).

8.2.1 Effects of different number of MFCCs

Using the same training data for the background model as in our original experiments, ten 300-segment UBMs were trained on different sets of MFCCs.

In each set a different number of MFCCs (from 1 to 10) were used, including MFCC_0. For example, in the first set only MFCC_0 was used; in the second set MFCC_0 and MFCC_1 were used; and in the tenth set MFCC_0 and MFCC_1 to 9 were used.

Figure 8.1 shows the distribution of MFCC_0 slopes of UBMs with different feature vector dimensions. To make a clear view only 5 of the 10 sets were showed on the graph. It can clearly be seen that as the number of MFCCs increases, the percentage of non-zero slopes decreases. The same tendency appears to the other MFCC channels as well. See Appendix D for the full graph.

This suggests that the lack of non-zero slopes is due to the maximum likelihood training algorithm giving priority to modelling stationary regions, together with the combinatorics of modelling these regions for all of the MFCC parameters. If this is the case (i.e. that all of the segments are "used-up" modelling stationary regions), one would expect to see more non-zero slope values if the number of segments is increased.

8.2.2 Effects of different number of UBM segments

The purpose of the second experiment in this chapter was to discover the effect of varying the number of segments on the trained UBM slopes. We fixed the number of MFCCs to six (MFCC 1 to 5 plus MFCC_0), and increased the number of segments in SHMMs from 300 to 2100, with the intervals equal to 300. The maximum duration τ_{max} was set to 50ms in this experiment.

Figure 8.2 shows the changes of MFCC_6 slopes. The MFCC_6 slopes of

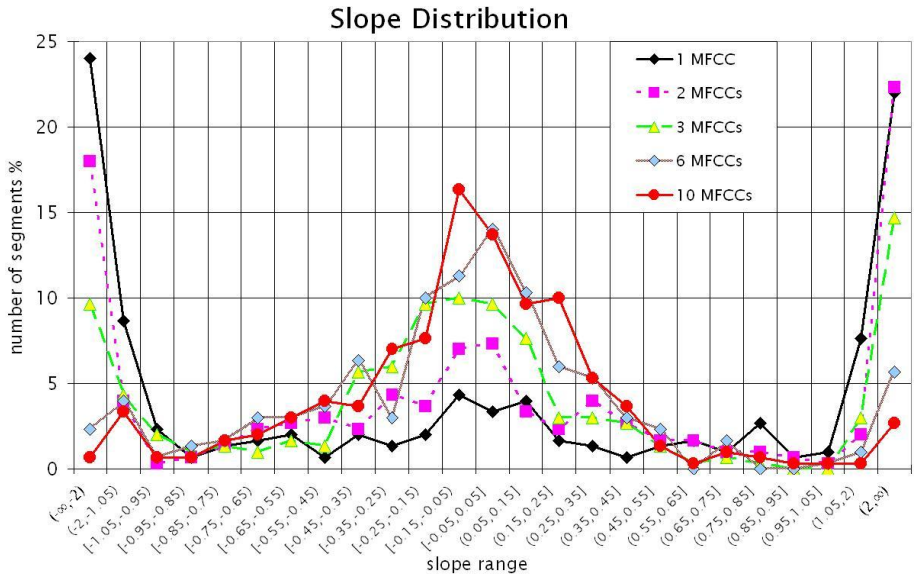


Figure 8.1: Distributions of slopes of MFCC_0 in the UBM as the number of MFCCs increases.

the UBMs with 300 and 2100 segments are drawn as solid curves with circle and square marks, separately. MFCC_6 slopes of other UBMs are shown as dashed curves [1].

As predicted as the number of segments increases, a larger percentage of segment trajectories tends to have bigger slopes. The percentage of segment trajectory slopes in the range around zero decreases as the total number of segments gets bigger from 300 to 1500. This confirms our theory that the maximum likelihood training priorly focuses on the stationary regions. When the number of segments increases more segments can be used to model the dynamic regions. As the total number of segments increases from 1500 to 2100, although a larger percentage of segment trajectories have bigger slopes, the percentage of segments whose slopes are just around zero (in the region

[1]MFCC_6 was chosen randomly. All the other MFC coefficients show the same tendency.

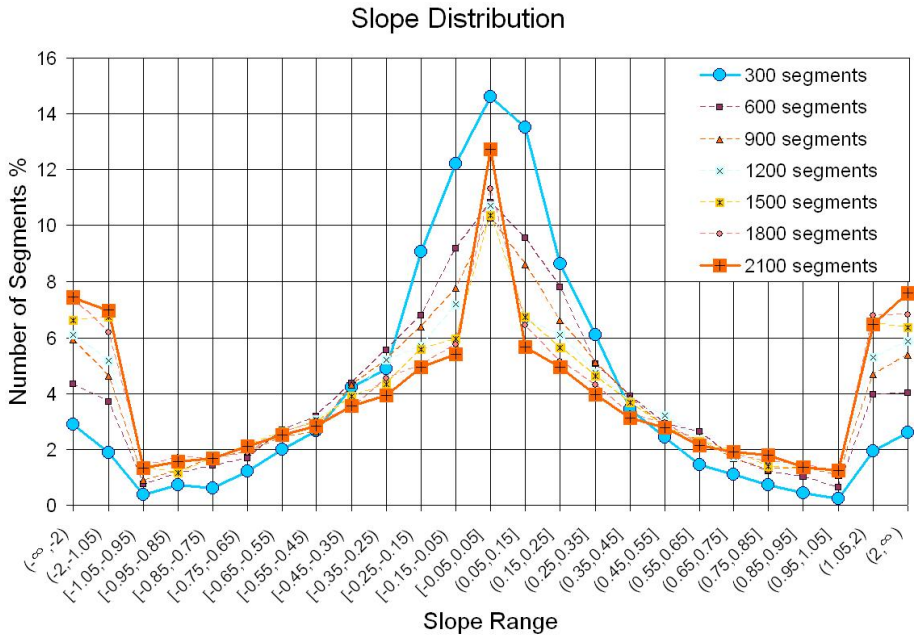


Figure 8.2: Slopes of MFCC_6 in UBM as number of segments increases.

$[-0.05, 0.05]$) increases again. This could be because at 2100 segments the number of segments is sufficiently abundant that there are enough segments to model the dynamic regions, hence more segments can again be used to model the stationary regions. This again shows that the priority of the ML training is to focus on the stationary regions.

8.3 Comparison of UBMs in GMM and SHMM system

These experiments demonstrate that as the number of MFCCs increases, or as the number of segments decreases, the system will have more zero-slope segments after maximum likelihood training. By contrast, as the number of MFCCs decreases, or as the number of segments increases, the system will

end up having more non-zero segment slopes. Without any supervision, the maximum likelihood training seems to give priority to model the stationary regions. When there are abundant segments in the system, more of them can be used to model dynamic regions. Or, when the system has a smaller number of MFCC channels, modelling dynamic regions in one MFCC channel does not conflict with the priority of modelling stationary regions in many other MFCC channels, more segments in the system can be used to model dynamic regions.

From the analysis of the above sections, it seems that it is the unsupervised maximum likelihood training which compromises the segmental trajectory structure, making it focus on the stationary regions, rather than the dynamic regions. Is this only a characteristic of our segmental system? We wanted to compare the dynamic information contained in our segmental GMMs with any dynamic information contained in conventional GMMs.

As previously stated, there are no differential parameters in our SHMM system because we hoped to represent acoustic dynamics by using segment trajectories. We constructed a 300-state SHMM system with a maximum duration length τ set to 2 (20ms). The segment slopes were then analyzed and compared with the 'delta' parameters in a traditional GMM-based system.

For the conventional GMM-based system trained on Switchboard we chose a 2048 component system built by Hansen et al. at the Air Force Research Laboratory (AFRL) at Wright-Patterson Air Force Base in Dayton, Ohio, USA (Hansen et al. 2004). The system was given to us by Eric Hansen. The equal error rate achieved in the 2003 NIST evaluations with this type of system by AFRL is around 5%. The AFRL system is based on

the MIT Lincoln Laboratory system. It uses the MIT Lincoln Laboratory speaker recognition system to extract MFCCs, RASTA filtering and energy based speech activity detection.

Figure 8.3 compares the distribution of slope values in our SHMM and delta values in the conventional GMM. Surprisingly, the delta parameters in GMMs are even smaller. 57.1% of the GMM delta parameters are distributed in the range around zero, compared with 28.9% of our SHMM trajectories. Although the AFRL model has 2048 components, much more than the 300 components which our segmental GMM has, the rest of the deltas of the AFRL model are still very close to zero. There are not much dynamics in the AFRL model.

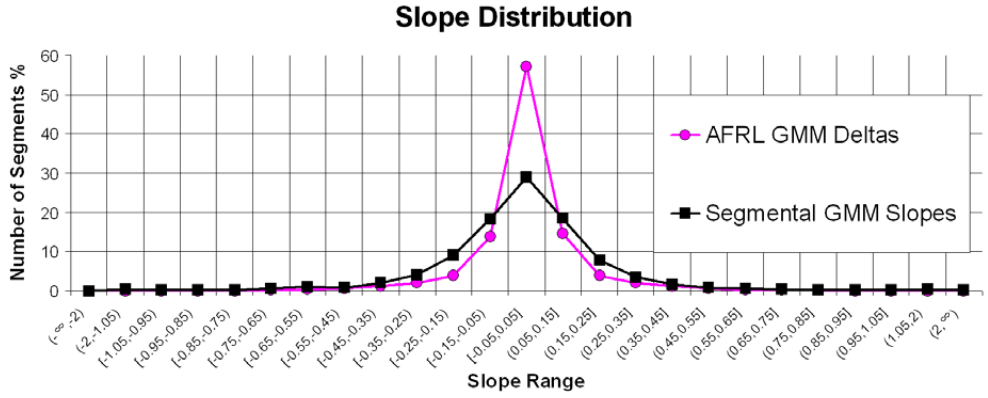


Figure 8.3: Statistics of GMM deltas and SHMM segment slopes.

The result indicates that the absence of a model of dynamics in TI-SV is not only a feature of our SHMM system. It also appears to be a feature of a conventional TI-SV GMM or at least the one provided by AFRL. The analysis of AFRL's model is consistent with our hypothesis. This evidence also suggests that the role of delta features in such a system is not to model

dynamics but to focus the modelling onto the stationary regions of a speech signal.

8.4 Summary of Analysis on TI-SV system

To explain the unexpected TI-SV results obtained using SHMMs on Switchboard data, some analyses have been carried out.

First of all, we have visualized the individual segment GMMs by applying an inverse Discrete Cosine Transform to the MFCC vectors. Although the visualized segments look all normal, most of the segment models correspond to quite flat trajectories. They do not seem to carry much dynamic information.

We then did experiments to measure the effects of different sets of MFCCs on the trained UBM. As the number of MFCCs increases, or , as the number of segments decreases, the percentage of non-zero slopes decreases. This suggests that the lack of non-zero slopes is due to the maximum likelihood training algorithm giving priority to modelling stationary regions.

We then compared our segmental models with a conventional GMM system, which was also trained on Switchboard. Analysis shows that more than half of the delta parameters in the conventional GMM background model are in the range of $[-0.05, 0.05]$, compared with nearly thirty percent of the SHMM trajectories. Thus, the role of the delta features in a conventional GMM system using maximum likelihood training seems to be to focus the system onto the stationary region of the speech, rather than to represent the dynamic regions. In other words, in order to secure a high probability

with respect to a given component, an acoustic vector not only needs to be close to the component mean, it must also be in a stationary part of the speech signal so that its delta parameters are close to zero(see the HMM assumptions illustrated in Figure 5.2). Although this is not something which is optimized directly in ML training, but ML training tries to maximize the probability of the training data and this seems to be achieved by focusing on stationary regions and hence having zero deltas. A consequence of this is that in recognition, the signal which match the segment means but don't have zero slopes won't get high probabilities.

The differential parameters are usually seen as dynamic features that are a measure of the change in the static features. In conventional HMM and GMM systems, it is assumed that each observation is static, and there is no dependency between the observations. By augmenting the original set of static acoustic features with differential features, the correlations between each observation and its neighbors can be captured to some extent, as well as the local dynamics in the speech. However, if most of the differential parameters are close to zero, their role seems to be to lay a strong emphasis on the static regions, and to diminish the correlation between neighboring observations.

A question arises as whether this is also the case for the TD-SV experiments on YOHO using SHMMs. The TD-SV results on YOHO show improvement on performance by applying segmental HMMs. Although both TD- and TI-SV use maximum likelihood training, in supervised TD-SV on YOHO we built models of labeled triphones, while in unsupervised TI-SV we build models representing unknown phoneme-like units. The supervised

ML training should help the system exploit dynamic information.

We wanted to find evidence that the SHMMs trained on YOHO contain dynamic information, and it is these dynamic information which lead to the improvement of speaker verification performance. Next chapter presents analyses on the YOHO TD-SV system.

CHAPTER 9

Analysis of Text-Dependent Speaker Verification system

In text-dependent speaker verification the speech is transcribed. This means that the training can be supervised and that a prescribed set of models is required to model a particular piece of speech. Potentially this forces the models to take account of non-stationary regions in the speech signal. In Chapter 6 the SV results we produced by applying SHMMs show an improvement. In this Chapter we return to the YOHO results to look for evidence that the improvement on TD-SV scores is due to better representation of dynamics by SHMMs.

Firstly we investigate the issue that the number of parameters is different in the SHMM and HMM systems. Then we examine the SHMM slope parameters in the YOHO system to see if they contain dynamic information. We also investigate whether there is a link between the segment models which have bigger slopes and better speaker verification scores. Finally a GMM

system has been built on YOHO to investigate whether the TI-SV system can capture some speech dynamics on YOHO (instead of Switchboard).

9.1 An HMM system with static and delta MFCCs

As mentioned in Chapter 6, we didn't use the Δ or Δ^2 parameters in our HMM and SHMM models. A side effect of this is that the number of parameters in the SHMM system is greater than the conventional HMM system. So the SHMM system has an unfair advantage here. We then built another conventional HMM system which has both MFCC statics and deltas. This static-plus-delta system was built using the same system settings as the conventional HMM system which does not have deltas in it. The performance of this system, as well as the performances of the SHMM system and the HMM system which only have MFCC statics (the latter two were shown in Figure 6.2), are shown in figure 9.1.

The results show that the conventional HMM system using both MFCC statics and deltas (the black curve) outperformed the segmental HMM system (the red curve). We believe that this is due to the advantage of the conventional HMM system that it has more parameters than the segmental HMM system does. But then the question arises of whether the better performance of the SHMM system, compared to the HMM system which has only MFCC statics, is due to the advantage that it also has more parameters than the HMM system. We then did some analysis on the different number

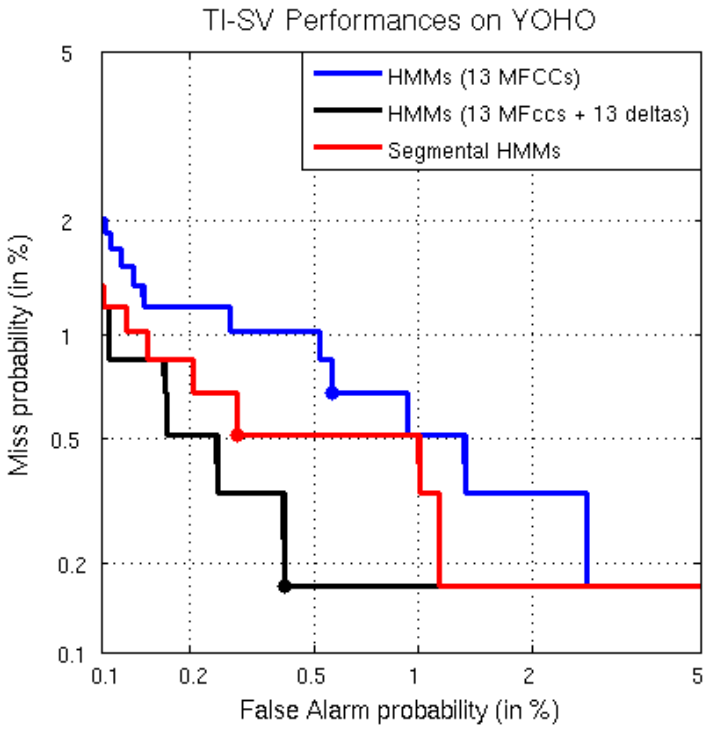


Figure 9.1: Results of the HMM and SHMM systems on YOHO.
blue curve: conventional HMM using 13 MFCCs
red curve: Segmental HMM using 13 MFCCs
black curve: conventional HMM using 13 MFCCs plus 13 deltas

of parameters between the HMM and SHMM systems.

9.2 Analysis on the number of parameters

In a conventional HMM system, the parameters involved in each state of the model are means of the extracted feature vectors (MFCC statics in our case),and the time derivatives of the feature vectors (the delta parameters, or sometimes the delta and acceleration parameters), the variances of the above parameters, the transition probabilities and a duration parameter which can be derived using the self transition probabilities of each state. If there are more than one stream or more than one Gaussian mixture component in a state, then there are separate means and variances for each component, and the weights of the streams or components are involved too.

In a segmental HMM system however, the parameters involved in each segment of the model are the parameters of the trajectories. These parameters are the midpoints, the slopes and the duration probabilities of the trajectories. As in an HMM system, there are also the transition probabilities between states. However, because the SHMM system has a duration pdf, there is normally no self transitions of the states, except for in the silence model.

So, in a conventional HMM system with only MFCC static parameters and single component Gaussian states, there are 13 MFCCs and 13 variances, plus the transition probabilities per state. In the corresponding conventional HMM system with MFCC statics and deltas, there are 13 MFCCs, 13 deltas and 26 variances, plus the transition probabilities. In a segmental HMM

Table 9.1: Number of parameters in HMM and SHMM systems

number of parameters	HMMs (static MFCCs only)	HMMs (static + delta MFCCs)	SHMMs
means (per state per GMM component)	13	26	26
variances (per state per GMM component)	13	26	13
transition probabilities (per model)	9	9	6
duration probabilities (per state)	0	0	15
total parameters (per model, assuming 3 states per model)	26*3+9=87	52*3+9=165	39*3+6+15*3 =168
total non-duration parameters (per model)	87	165	123

system, there are 13 midpoints, 13 slopes and 13 variances, plus the transition probabilities and the duration probabilities. The numbers of different parameters of these three systems are compared in Table 9.1. As there are same numbers of states in all three systems, we are comparing the number of parameters in each state.

The other parameters of these systems which include the transition probabilities and the duration probabilities are also compared in Table 9.1. In a conventional HMM system which has 3 emitting states connected in a left-to-right manner, there are 9 parameters including 3 self transition probabilities.

The self transition probabilities defines the geometric duration probabilities of each state. In a SHMM system with also 3 left-to-right emitting states there are only 6 transition parameters. In a SHMM system there are also the duration probabilities how many of which are decided by the maximum duration set in the system. For the YOHO SHMM system the maximum duration is 15 so there are 15 duration probabilities in the system. In a conventional HMM system there is no separate duration parameter. The geometric duration probabilities can be calculated using the transition probabilities.

Table 9.1 does not distinguish between different types of parameters, simply giving an overall total. However, it is well known that all parameters are not equal. For example, variance parameters require more training materials than the corresponding mean parameters. In the past, this has motivated approaches such as "grand variance", where all PDFs share the same variance (Russell and Ponting 1990) or "tied variance" where sets of PDFs share the same variance (Young 1992). In early work of HMMs it was also noted that the state transition probabilities contribute less to recognition (Juang and Rabiner 1991), and the same is true of duration parameters in general (Juang and Rabiner 1991). As in the SHMM system the number of duration parameters is outstanding compared to the HMM systems, the last row of Table 9.1 also gives a total non-duration parameters. In summary, the simple totals in table 9.1 do not tell the whole story. In the next section we look at varying the numbers of some of these parameters, and the effects on accuracy.

9.2.1 Experiments to reduce the number of parameters of the SHMM system

We can not build an SHMM system which has exactly the same number of parameters as the HMM system with static and delta MFCCs. We can not build an HMM system which has exactly the same number of parameters as the SHMM system either. So we try to reduce the parameters in our SHMM system to make it comparable to the HMM system with only static MFCCs. As the slopes are the extra parameters of our SHMM system compared to the HMM system, the slopes can be set to zero to make the two systems more equivalent. The number of duration parameters of the SHMM system is also more than the number of duration parameters of the HMM system. There are 15 duration parameters in the SHMM system, while in the HMM system the duration has a geometric pdf with a single parameter. To make the two systems have closer to the same number of parameters, we need to use the same duration pdf for both systems. Hence we adjusted the parameter settings in the SHMM system and built five SHMM systems as following:

- System A: zero trajectory slopes and geometric duration pdfs
- System B: zero trajectory slopes and uniform duration pdfs
- System C: non-zero trajectory slopes and geometric duration pdfs
- System D: non-zero trajectory slopes and uniform duration pdfs
- System E: non-zero trajectory slopes and 15 duration probability parameters

In Systems A,B,C and D the duration probabilities of each state are calculated using the self transition probabilities from their counterpart states in the HMM system. In Systems A and B the trajectory slopes are set to zero so that the two systems have exactly the same number of parameters as the conventional HMM system with only static MFCCs. System A has a geometrically distributed duration pdf, the same as in the HMM system. Thus the only difference between System A and the HMM system (with only static MFCCs) is that there is a maximum duration limit which is set to 15 in the SHMM system, while in the HMM system there is no such limit. System A is to be compared directly with the HMM system with only static MFCCs. System B has a uniformly distributed duration pdf. This is to see whether the details of different duration pdfs affect the system performances. System C and D both have non-zero trajectory slopes, and they have a geometric and uniform duration pdfs respectively. System E is a standard segmental HMM system, with non-zero trajectory slopes and non-parametric duration probabilities calculated from the training data. The number of parameters in these five systems are given in Table 9.2.

The results of the five systems and the static-only HMM system are shown in figure 9.2. As we can see from the results, both System A and B achieve slightly worse SV results than the conventional HMM system with static-only MFCCs. The maximum duration setting (15 frames) in system A may have limited the model from accommodating longer segments which may have appeared in the data, and lead to a poorer performance. The setting 15 came from experiments on TIMIT (Jackson and Russell 2002), which showed that it is a reasonable number for ASR and SV experiments on TIMIT. But

Table 9.2: Number of parameters in HMM and SHMM systems

number of parameters	System A	System B	System C	System D	System E
means (per state per GMM component)	13	13	26	26	26
variances (per state per GMM component)	13	13	13	13	13
transition probabilities (per model)	6	6	6	6	6
duration probabilities (per state)	1	1	1	1	15
total parameters (per model, assuming 3 states per model)	26*3+6 +1=85	26*3+6 +1=85	39*3+6 +1=124	39*3+6 +1=124	39*3+6 +15*3 =168
total non-duration parameters (per model)	84	84	123	123	123

it may be not big enough for YOHO.

The DET curves of System A and B are almost identical. So are the DET curves of System C and D. The results suggest that the difference between the geometric or uniform duration pdf doesn't make a difference to the SV performance. This could be because that the duration pdfs can be easily swamped by the Gaussian pdfs of the model. In other words, the difference between duration probabilities for different segment lengths will be dwarfed by differences between the "acoustic segment" probabilities. System E, different from System C and D, uses a non-parametric duration model. There

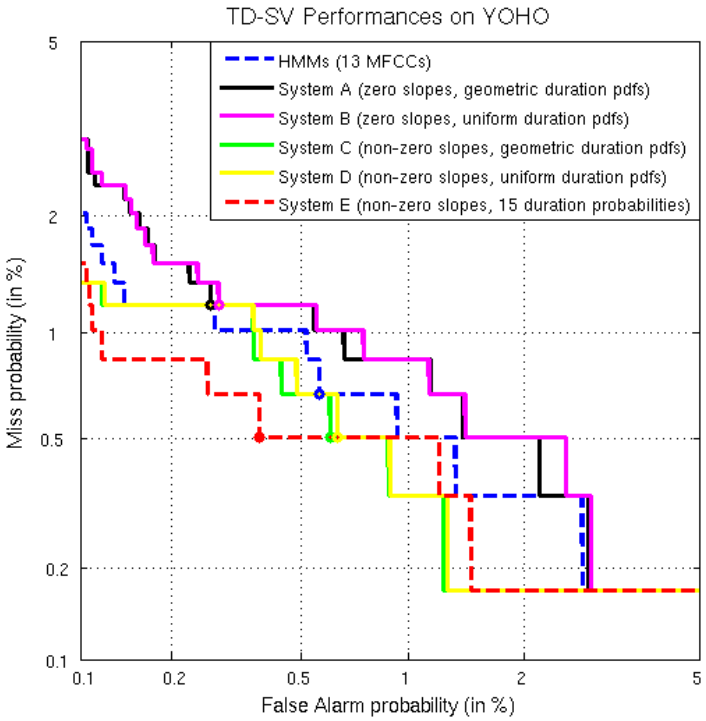


Figure 9.2: TD-SV results.
blue dotted curve: HMM system with MFCCs and no deltas
black curve: System A (SHMM system with zero trajectory slopes and geometric duration pdfs)
magenta curve: System B (SHMM system with zero trajectory slopes and uniform duration pdfs)
green curve: System C (SHMM system with non-zero trajectory slopes and geometric duration pdfs)
yellow curve: System D (SHMM system with non-zero trajectory slopes and uniform duration pdfs)
red dotted curve: System E (SHMM system with non-zero trajectory slopes and 15 duration parameters)

are zero or very small duration probabilities which cannot be left out during the probability calculation. The issue is that in System C and D the differences between different duration probabilities are small (zero is the case of System E), but for System E these differences can be arbitrary, and the better performance indicates that they can actually influence the result. All three systems are confined by the same maximum duration setting. The maximum duration setting can stop the segmental model from having long segments and hence affect the SV performance. For YOHO we have not tested the SHMM system to get an optimal maximum duration setting because increasing the maximum duration setting involves larger computational load and longer system running cycle (see 7.3 about the computational load).

9.2.2 Different ways of using parameters between the systems

Referring to Figure 5.1, the SHMM system assigns a high probability to data if it lies inside a "tube", whose width is determined by the variance parameter, centred on the trajectory. There is no constraint on the "local" slope values within this tube (as illustrated in Figure 5.1). By contrast, in a conventional HMM with static and dynamic parameters, the local dynamics must match the state slope throughout a segment if a large probability is to be achieved.

Based on the above results, the performance of a system can not be simply judged by the different number of parameters. The HMM and SHMM systems are two totally different systems and each has its own ways of using

their parameters. We can not make an HMM system have the same number of parameters as the SHMM system with non-zero slopes. Which 13 parameters should we choose to add to each state of the HMM system? The above results show a comparison between a deteriorated SHMM system and the HMM system with only static MFCCs, in which the HMM system outperform the deteriorated SHMM system. The different duration pdfs of the two systems, the maximum duration set in the SHMM systems, and maybe other unknown factors could all affect the SV performances.

The application of the SHMM system on SV is to see whether it can catch speech dynamics and use the dynamic information to improve SV performance. Some analysis are carried out in the following sections to investigate this issue.

9.3 Analysis of SHMM slopes in TD-SV system

To see whether the SHMM trajectories represent speech dynamics, firstly, we examined the SHMM slope parameters. The statistics of UBM slopes show that the YOHO SHMM slopes are more diverse than the Switchboard system. Figure 9.3 compares the distribution of the trajectory slope values in the cases where the 'background' model is a 'segmental GMM' (used for TI-SV on Switchboard), phone-level SHMM (used for TD-SV on YOHO), and the distribution of the delta values in a conventional 'background' GMM (again used for TI-SV on Switchboard).

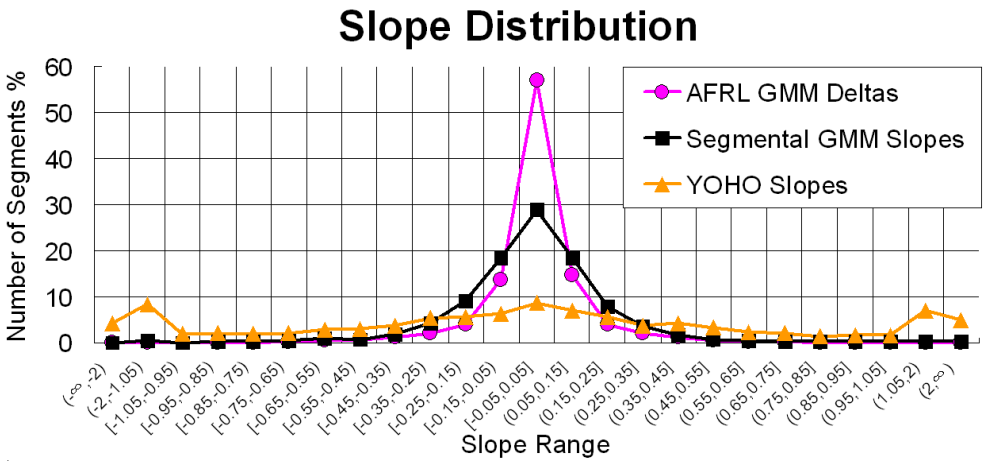


Figure 9.3: Distribution of deltas and segment slopes of three systems.
violet curve: AFRL GMM deltas for TI-SV on Switchboard
black curve: Segmental GMM segment slopes for TI-SV on Switchboard
orange curve: SHMM segment slopes for TD-SV on YOHO

The figure illustrates that the slope values for the two TI-SV models are concentrated around zero, while a larger proportion of the slope values in the TD-SV models are significantly non-zero. These bigger slope values indicate a greater emphasis on modelling speech dynamics. The YOHO phone-SHMM slopes are most diverse of the three. Less than 9% of the segment slopes are distributed in the range around zero. Compared to the other two systems, the YOHO SHMMs manage to model more dynamic information.

The comparison between the slope/delta values of these three systems should ideally be performed on the SDMs. However we do not have the SDMs of the AFRL GMM system. And there is also the issue that the speakers in the YOHO and Switchboard systems are different. If we want to compare them we have to take an average of the slope/delta values of all the SDMs and that will perhaps be very close to the slope/deltas of the UBMs. Nevertheless, we know that in these three systems the SDMs are

all adapted from the UBMs using MAP adaptation and a sparse amount of data. Many parameters of the SDMs after adaptation will remain the same as those of the UBMs due to not having enough training materials. Thus the slope/delta distributions of the SDMs should not be far away from the ones of the UBMs. Figure 9.4 shows the segment slope distributions of both the UBM and the SDM of speaker 101 in the YOHO TD-SV system. The SDM of speaker 101 has slightly bigger slope values than the UBM. But the slope distribution curve of the SDM is still very close to the slope distribution curve of the UBM. The slope/delta distributions in Figure 9.3 gives us a clear comparison of the amount of dynamic information captured between the three systems. The YOHO SHMMs model more speech dynamics than the other two systems.

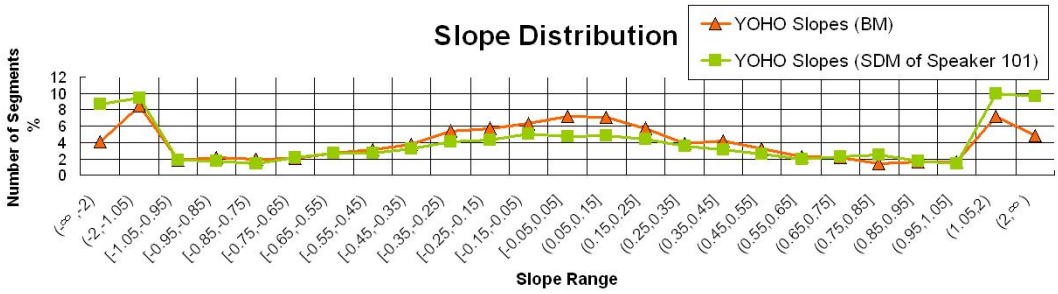


Figure 9.4: Distribution of segment slopes of the YOHO TD-SV system.
green curve: SHMM segment slopes of speaker 101's SDM
orange curve: SHMM segment slopes of the UBM

9.4 Relationships between the SHMM trajectory slopes and the SV scores

Do these dynamic regions that the YOHO SHMMs model contribute to speaker-verification accuracy? Experiments were conducted to find out if there is any relationship between the contribution to the YOHO speaker verification score due to a segment and the SHMM trajectory slopes for that segment. By measuring the likelihood ratio $p(Y\ SDM)/p(Y\ UBM)$ for individual segment $Y_{t_n} = \left[y_{t_{n-1}+1}, y_{t_{n-1}+2}, ..., y_{t_n}\right]$ of a speech signal, we could find out the relative contributions of static and dynamic segments to the speaker-verification decision.

For a test utterance $Y = \left[y_{t_0+1}^{t_1}, y_{t_1+1}^{t_2}, ..., y_{t_{N-1}+1}^{t_N}\right]$, the speaker verification score is computed by

$$L(Y) = \frac{\prod_{n=0}^{N} P(y_{t_n+1}^{t_{n+1}}\ s_n, SDM)}{\prod_{n=0}^{N} P(y_{t_n+1}^{t_{n+1}}\ s_n, UBM)}, \tag{9.1}$$

where s_n is the n^{th} segment, N is the number of segments in the segment sequence which have generated the observations. Because the same state (segment) sequence was used for the UBM and the SDM, Equation 9.1 becomes

$$L(Y) = \prod_{n=0}^{N} \frac{P(y_{t_n+1}^{t_{n+1}}\ s_n, SDM)}{P(y_{t_n+1}^{t_{n+1}}\ s_n, UBM)}. \tag{9.2}$$

Hence the term

$$\frac{P(y_{t_n+1}^{t_{n+1}}\ s_n, SDM)}{P(y_{t_n+1}^{t_{n+1}}\ s_n, UBM)}$$

is a measure of the contribution of state s_n to the speaker verification score.

Linking the segment level score with the UBM segment slopes can show any relationship between the contribution of state s_n and the dynamic information exploited in this state.

The segment-level scores were extracted and compared with the UBM segmental trajectory slopes (Figure 9.5). The scores are the average scores for each segment over all test samples of this segment. In total 127 context-sensitive triphone SHMM states (from 43 triphone models) were used in the YOHO TD-SV system. All SDM scores have been normalized by the UBM scores in the logarithmic domain and normalized by segment durations. The sum of all 13 MFCC slopes (absolute values) in each segment was calculated to show the "non-stationarity" of each segment. The bigger the true speaker scores or the smaller the impostor scores, the better the contribution of the segment to speaker verification.

A baseline system was also built for a reference (Figure 9.6). In the baseline system all the triphone models were trained exactly the same way except that the slopes in the models were set to zero and not reestimated during training. Verification tests were also performed on these models and the normalized segment-level scores were extracted. Although in this baseline system all the segments have a zero slope, to make a clear comparison with the system with non-zero slopes, Figure 9.6 uses the same slope distribution of UBM segments as in Figure 9.5 to locate the speaker verification scores.

Inspection of Figure 9.6 suggests that in the baseline system the distribution of the true speaker scores and the impostor scores largely overlaps. Compared to Figure 9.6, in Figure 9.5 the true speaker scores and the impostor scores are more separately distributed. For the system with nonzero-

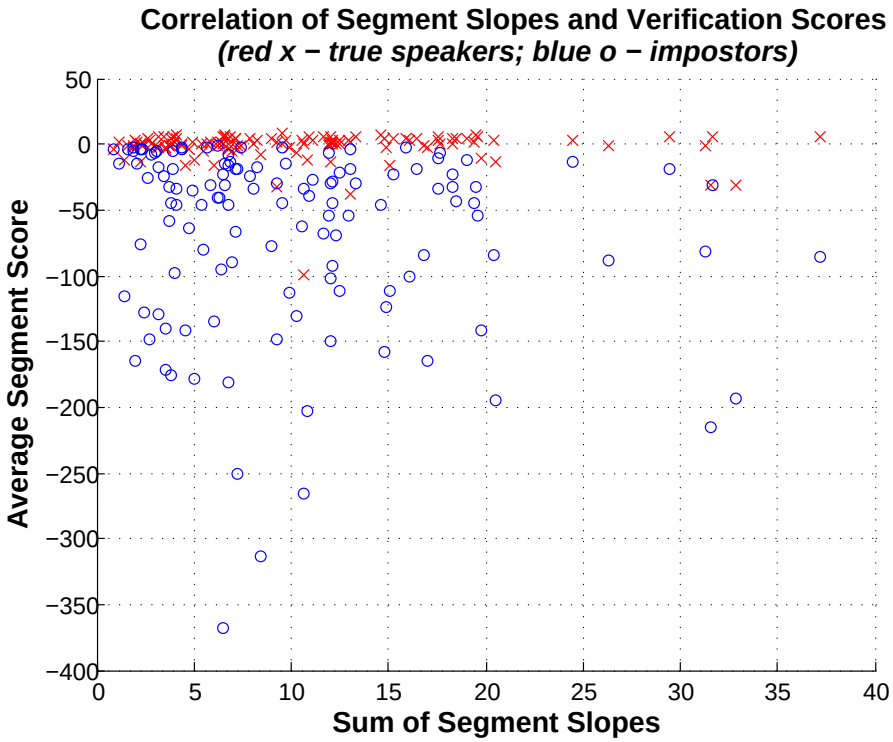


Figure 9.5: Relationship between SHMM segment slopes and TD-SV scores.

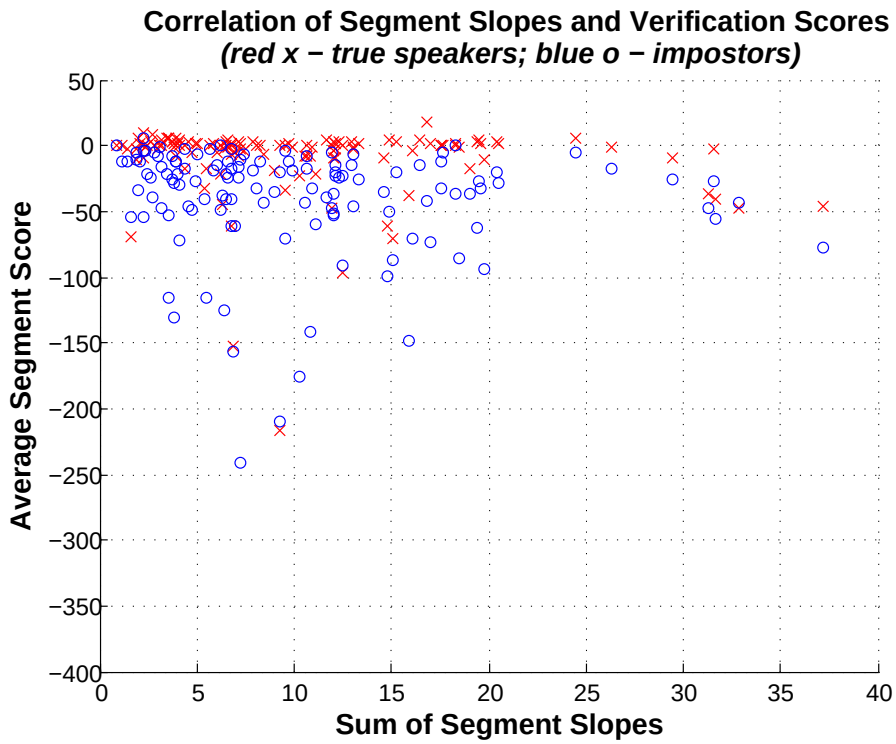


Figure 9.6: Relationship between the zero-slope SHMM segments and TD-SV scores. *Uses the slope distribution of SHMMs with non-zero slopes for comparison.*

slope segments, as the segment slope increases, the segment-level true speaker scores vary little around zero. When compared to the zero-slope models, the nonzero-slope models enhance the true speaker scores significantly especially in the slope range from five to fifteen. Also as the segment slope increases the nonzero-slope models make the impostor scores dramatically worse.

An interesting discover in comparing Figure 9.5 and Figure 9.6 is that both systems have a similar score distribution on the slope scales. For example, it seems that both system produce small impostor scores in the slope range between 5 and 15. Considering that both systems start with the same initial parameter values before being trained with different slope setting, this shows strong connection between the corresponding mixture components in the zero-slope and nonzero-slope systems.

Figure 9.7 shows clearly the comparison of both systems. Instead of showing all the scores for each segment, the average scores of all segments from each slope range, $(0, 5], (5, 10], ...(35, 40]$, are calculated. The trajectory slope SHMMs were represented as the solid line. The zero-slope SHMMs were represented as the dashed line. The number of segments in each slope range was also displayed.

The analysis shows that the nonzero-slope segments have bigger true speaker scores and smaller impostor scores. The increases of true speaker scores are most significant in the slope range from five to twenty and the decreases of impostor scores are most significant in the slope range from zero to fifteen. Both areas contain most of the segments. If we choose the intersection of the two areas, which is between five and fifteen, divide it by thirteen, which is the dimension of the MFCCs, it gives us an average slope

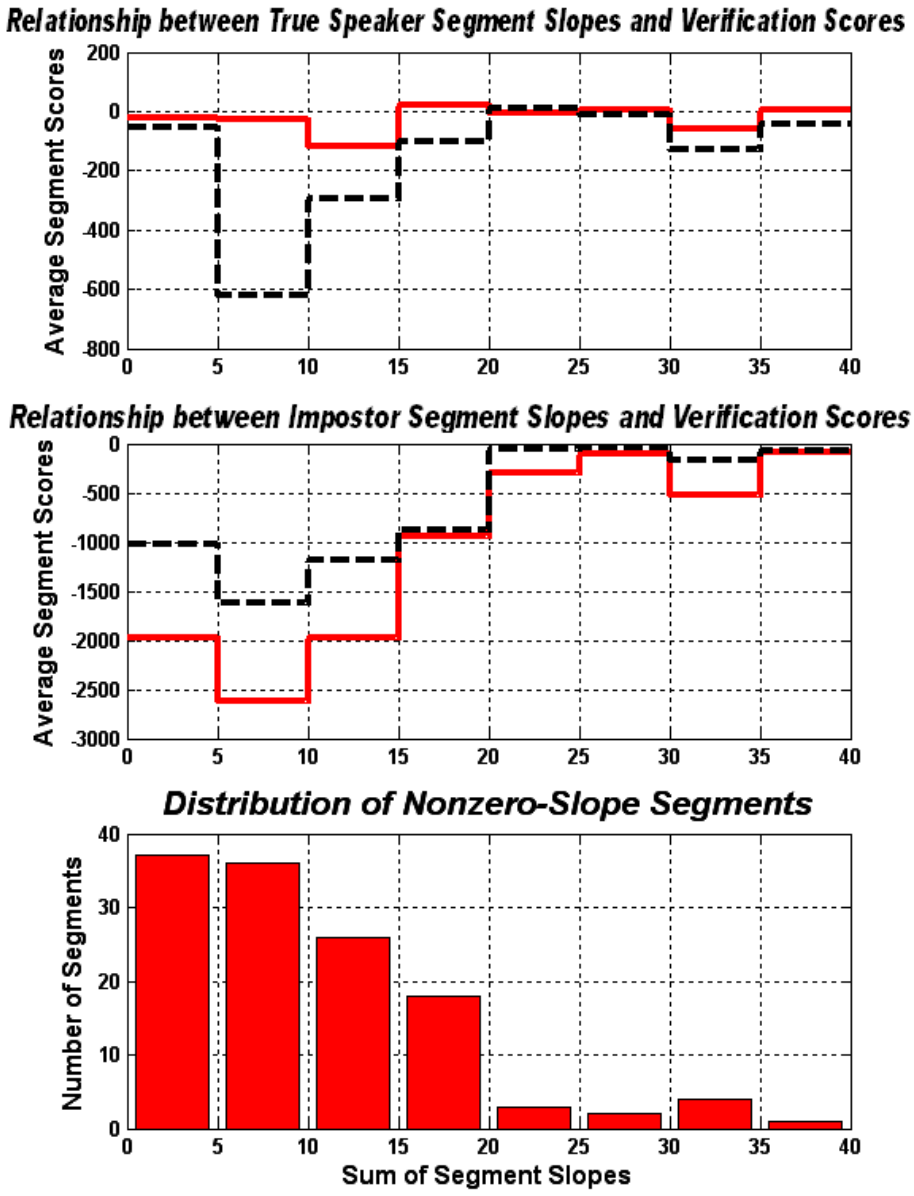


Figure 9.7: Comparison between the TD-SV scores of the nonzero and zero -slope segments (*solid line - trajectory slope SHMMs; dashed line - zero slope SHMMs).*

for each of the MFCCs, which is between 0.38 and 1.15. This MFCC slope range is very important for speaker verification according to the analysis, as a big percentage of the segments falls in this range, and these segments contribute more to speaker verification accuracy.

If we have a look at figure 9.3 again, which is modified and shown as figure 9.8, we can see that in the AFRL system and our segmental GMM system most deltas or trajectory slopes have an absolute value which is smaller than 0.38. If some of the states or segments in these systems can be used to model the important dynamics which is in the range of $[0.38, 1.15]$, it is hopeful that the speaker verification performance can be improved.

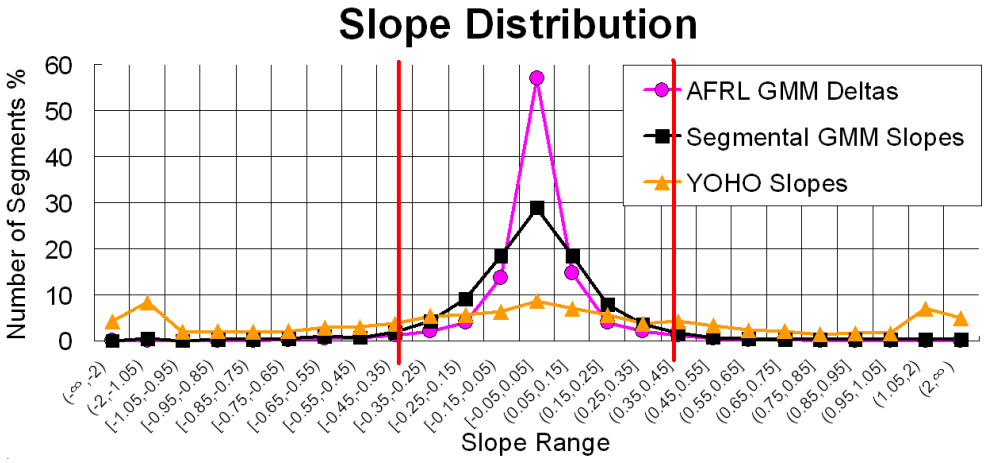


Figure 9.8: Distribution of deltas and segment slopes of three systems.
violet curve: AFRL GMM deltas for TI-SV on Switchboard
black curve: Segmental GMM segment slopes for TI-SV on Switchboard
orange curve: SHMM segment slopes for TD-SV on YOHO
red solid lines: where the slope values are -0.38 and 0.38

Table 9.3: Systems and databases

	Clean Data	Noisy Data
HMM	YOHO	unpractical
GMM	YOHO	Switchboard

9.5 GMM experiments on YOHO

HMMs were used for YOHO, a database which only has low level office noise, while GMMs were used for Switchboard, a database which contains much higher level noise and distortion (see 2.3). Direct comparison between the two systems is difficult due to the fact that the two databases are so different. To get a fair comparison, either a GMM system should be built on YOHO, or an HMM system can be built on Switchboard (see Table 9.3). Due to the huge computation load, the running time for the experiments on Switchboard is sufficiently long that it is unpractical to build an HMM system on Switchboard and try different system settings. Instead a GMM system was built on YOHO, so that the GMM system and the HMM system on YOHO can be compared directly.

To build a text-independent SV system on YOHO using GMMs, the YOHO data were treated as data without transcriptions. As in the conventional HMM experiments, 13 MFCCs plus 13 deltas were extracted. Exactly the same as in the HMM experiments, firstly a UBM was constructed using the material of 20 randomly chosen speakers. Then 118 SDMs were trained each using their own training material. GMM systems with different numbers of components were built and tested on the YOHO test material. The

results of the GMM systems are shown in figure 9.9.

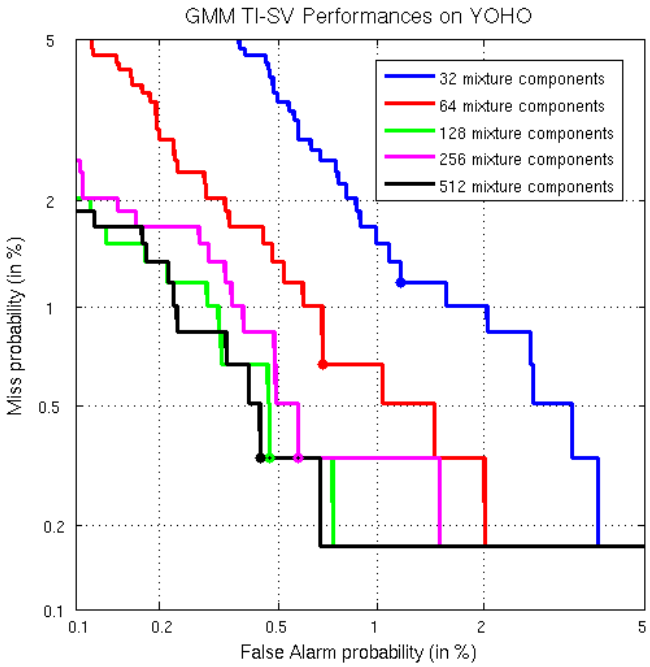


Figure 9.9: Results of the GMM TI-SV systems with different number of components.

The results show that the best performances are of the GMM systems with 128 and 512 mixture components. It seems that for the YOHO database, anything more than 128 components are not necessary as there are not enough data to make a model with a large number of components well trained. The system with 256 components even gives a performance worse than the system with 128 components. According to the number of parameters, the GMM system with 128 mixture components is the closest to our segmental HMM system. In the conventional HMM system and the segmental HMM system, there are 46 physical triphones, which have 136 states. The comparison between the GMM system with 128 components, the segmental HMM system,

and the two conventional HMM systems is shown in figure 9.10.

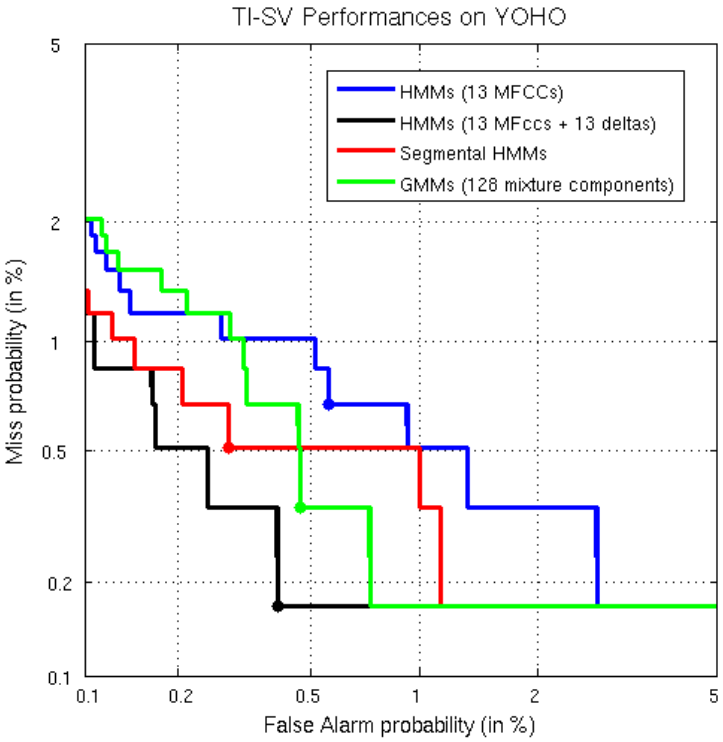


Figure 9.10: Result of the GMM TI-SV system on YOHO.
blue curve: conventional HMM using 13 MFCCs
green curve: GMM using 128 mixture components
red curve: Segmental HMM using 13 MFCCs
black curve: conventional HMM using 13 MFCCs plus 13 deltas

The results show that the GMM system (the green curve) performs similarly to the segmental HMM system (the red curve). The performances of both systems are worse than the HMM system using both MFCC statics and deltas (the black curve). It is worth noticing that the GMM system with 128 components has similar number of parameters as the segmental HMM system and the HMM system using both static and delta MFCCs (Table 9.4). However, the GMM system and the SHMM system do not seem to perform

Table 9.4: Number of parameters in HMM, SHMM and GMM systems

number of parameters	HMMs (static MFCCs only)	HMMs (static + delta MFCCs)	SHMMs	GMMs (128 mixture components)
means (per state per GMM component)	13	26	26	26
variances (per state per GMM component)	13	26	13	26
transition probabilities (per model)	9	9	6	257
duration probabilities (per state)	0	0	15	0
total parameters (of all models)	(26*3+9) *46 =4002	(52*3+9) *46 =7590	(39*3+6 +15*3)*46 =7728	52*128+257 =6913

as good as the HMM system, which again confirms that the performances of different systems can not be simply judged by the number of parameters in these systems.

I then had a look at the slope/delta distributions of the three systems. The comparison of their slope/delta distributions are shown in figure 9.11. The AFRL GMM system and the segmental HMM system, both of which were trained on Switchboard are also shown. As we can see from the graph, the distributions of the three systems on YOHO are very similar. This demonstrated that the dynamics in YOHO can be modeled using both HMMs/SHMMs for TD-SV and GMMs for TI-SV. In TD-SV, obviously

building models of labeled triphones helps the system exploit dynamic information. In TI-SV, the GMM system was very well trained on YOHO, due to the fact that YOHO has a very limited vocabulary (18 words) and is high quality data. When there are enough states to model the data, it seems that the dynamics can be captured accurately.

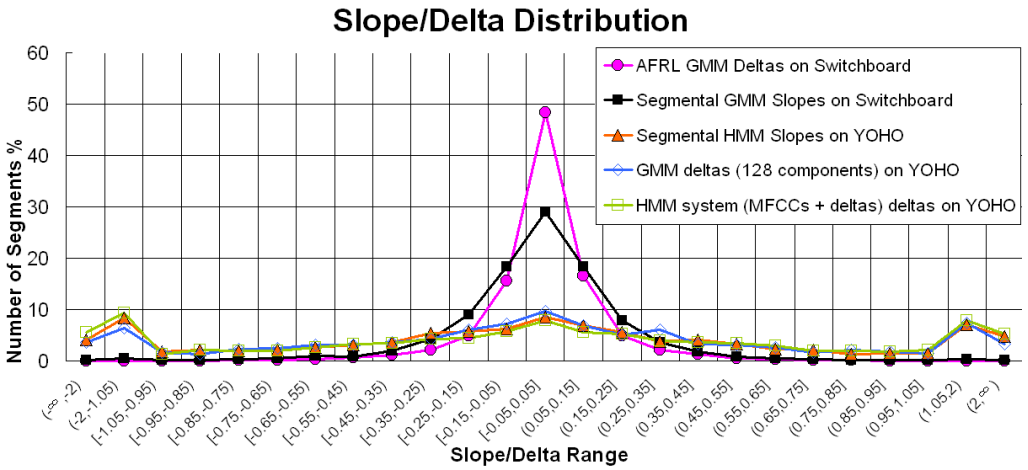


Figure 9.11: Delta/Slope distributions of the systems on YOHO.
magenta curve: AFRL GMM system on Switchboard
black curve: segmental HMM system on Switchboard
blue curve: GMM using 128 mixture components on YOHO
orange curve: Segmental HMM using 13 MFCCs on YOHO
green curve: conventional HMM using 13 MFCCs plus 13 deltas on YOHO

The delta distributions of the GMM systems with different number of mixture components are shown in figure 9.12. For a more clear view, the cumulative delta distributions are shown in figure 9.12. It is evident from the graph that as the number of mixture components increases, a larger percentage of deltas is used to model dynamics. This echoes the analysis of the segmental GMMs in 8.2, which again suggests that the priority of the maximum likelihood algorithm is to focus on the stationary region, but when

there are enough states in the system they start to model speech dynamics.

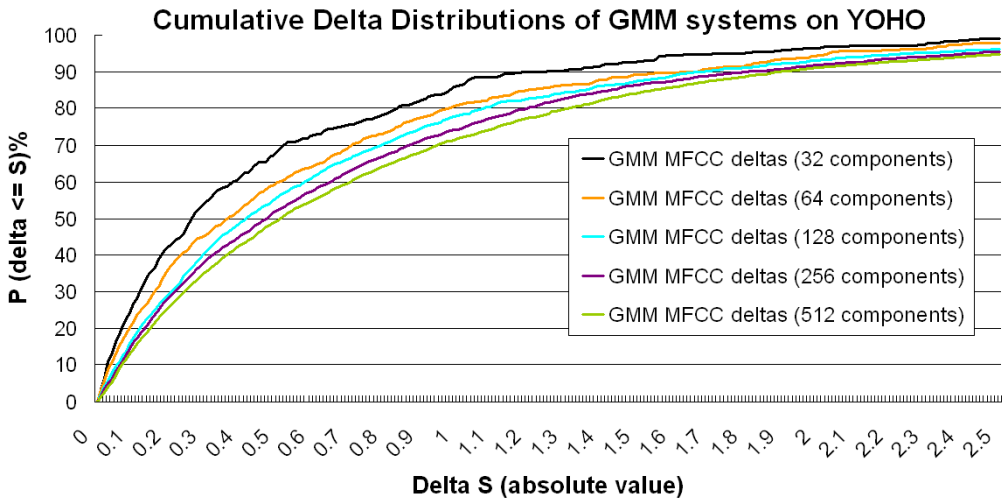


Figure 9.12: Cumulative delta distributions of the GMM systems on YOHO.

So, we have shown that for a GMM system, if there are enough mixture components then dynamics will be modeled. For YOHO, because it is simple, "enough" might be quite small, but for the additional complexity of Switchboard, many more components appear to be needed before modeling of dynamics begins. However, for a complicated database such as Switchboard which has a large vocabulary and noisy data, choosing the number of states for the system is not the more the better. The number of parameters we choose needs to depend on how much training data we have and using the available training materials how many of these parameters could be well trained.

9.6 Summary

An HMM system with both MFCCs and deltas was built to be compared with the SHMM system and to help investigate the issue of different number of parameters in both systems. We concluded that the performance of a system can not be simply judged by the different number of parameters. The HMM and SHMM systems are two totally different systems and each has its own ways of using their parameters.

The statistics of the phone-level SHMM slopes in the YOHO TD-SV system demonstrate that the YOHO models manage to contain dynamic information. Compared to the GMMs and segmental GMMs trained on Switchboard, phone-level segmental HMMs trained on YOHO have much bigger slope values.

Investigation of the relationship between the SHMM trajectory slopes and the SV scores unveils that most of the segments in the nonzero-slope models produce bigger segment-level true speaker scores and smaller segment-level impostor scores, and hence contribute to increasing the speaker verification performance. Thus, the SHMMs in a TD-SV system do contain speech dynamic information and from our analysis these dynamic regions do contribute to speaker verification accuracy. We have also demonstrated that the dynamics in the range of $[0.38, 1.15]$ are very important to speaker verification. From the analysis we can see that a large group of the segments are in this slope range, and these segments contribute significantly more to the speaker verification accuracy.

In TI-SV on Switchboard, however, we haven't seen the segmental GMM

exploiting much dynamic information. Neither have we seen any improvement of performance by using the segmental GMM. These results all suggest that the role of the differential parameters in unsupervised TI-SV is to give priority to the static regions, rather than to model dynamic regions.

A GMM TI-SV system was built on YOHO. In this case the GMM system manages to model the dynamics. We know that YOHO has a very limited vocabulary. Each word of the YOHO vocabulary has a sufficient number of high quality training samples. This helps the SV systems to accurately model the dynamics, provided that there is enough states to model the data. Switchboard, on the contrary, has a large vocabulary and quite noisy data. The complications of the Switchboard data, together with the priority of the ML training to model the static regions, make it a more difficult task to model the speech dynamics. The next chapter presents a more detailed study of this phenomenon using GMMs for TI-SV on the Switchboard corpus.

CHAPTER 10

TI-SV using Conventional GMMs

The majority of current GMM systems incorporate first-order derivative features, most often applied to a basic feature set of MFCCs and an energy feature, and many also include second-order derivatives. Most of the benefit from derivative features, as commonly believed, is due to their ability to capture dynamic information. These derivative features also have the useful property that they are not affected by any constant or slowly changing disturbances to the signal, which are additive in the feature domain, such as linear filtering in microphone pre-amplifiers and on telephone channels.

Our analyses of segmental GMM-based and conventional GMM-based TI-SV systems has shown that the delta parameters are useful not because they explicitly model dynamics but because they only give high probabilities to vectors which are close to the state mean and in stationary regions. Because all the ML training is concerned about is maximizing the probability, other factors are not considered during training. Of course, things might be different if the training scheme are some sort of discriminative training.

To investigate more thoroughly how traditional GMM systems handle dynamics and how ML training deals with 'delta' parameters in a GMM system, we built conventional GMM TI-SV systems each of which contain different feature sets. Previous research (Soong and Rosenberg 1988; Liu, He, and Palm 1996)[1] has concluded that using MFCC 'delta' parameters alone (i.e. no static parameters) in TI-SV leads to much poorer performance compared with either using static parameters alone or static plus 'delta' parameters. As our analyses are based on the Switchboard data, we want to see how the systems handle the delta parameters if we train our systems on these data. We build three traditional GMM systems using different parameter sets: sys_19 (19 static parameters (MFC_0 to MFC_18)), sys_19d (19 'deltas' (Δ MFC_0 to Δ MFC_18)), and sys_38sd (19 statics and their corresponding 'deltas'). We use the same Switchboard material that was used in our segmental GMM system (NIST SRE 2002 and NIST SRE 2003 data) to train and test these systems.

10.1 Experimental Methods

The experiments used HTK to build the UBM and SDM and to do the verifications. After the front-end processing the mel-frequency cepstral coefficients (MFCCs) were extracted. We applied Cepstral Mean Subtraction (CMS) over each speech utterance to remove possible convolutional noise due to channel effects. A simple energy-based speech-noise detector was used to judge which parts of the speech are noise and which are speech. Only when

[1]Soong and Rosenberg's work used a 10-speaker (5 male and 5 female), isolated digit databese. Liu, He, and Palm's study used the TIMIT corpus.

the energy of a state is higher than a pre-set signal/noise threshold the state is kept as speech, otherwise it is discarded as a silence-noise frame to remove irrelevant information. The signal/noise threshold was set to a conservative -6 (MFCC energy value) at this stage.

Three traditional GMM systems were built, one for each of the parameterizations: sys_19, sys_19d, and sys_38sd, . The background models were trained using the NIST 2002 SID one-speaker training material. Each UBM was initialized with a single mixture component with a global mean and variance. The model components were then repeatedly split (one to two, two to four, and so on) and reestimated until each of the GMMs contained 512 Gaussian mixture components. To avoid the occurrence of singularities with very small variances after iterative training, a variance floor is preset so that the variance is always larger than or equal to the given floor. When we increased the number of Gaussian mixture components to 1024, after re-estimation many components hit the minimum variance (i.e. diagonal element of the covariance matrix) limit which was set to 0.001 in HERest training tool. We concluded that 512 was the maximum number of components that could be well trained using the NIST SRE 2002 training set.

The Speaker Models were obtained from the background model by MAP adaptation (Lee et al. 1991) using the one-speaker data (207 females and 149 males) from the 2003 NIST SRE training set. During MAP adaptation, as the training data is comparatively sparse, for each component only the model means were re-estimated, according to Equation (10.1):

$$\bar{\mu} = r * \mu_S + (1 - r) * \mu_W, \tag{10.1}$$

where $\bar{\mu}$ is the new (MAP adapted) value of the mean, μ_S is the mean of the speaker adaptation data, μ_W is the mean from the UBM, and $r = n/(n+R)$, where n is the number of occurrences of the current component as the best scoring mixture component. The parameter R determines how many feature vectors from the adaptation set must be assigned to a mixture component for the adaptation data and the prior to contribute equally to $\bar{\mu}$, and is typically chosen empirically. Following Reynolds' work (Reynolds et al. 2000), R was set to 16. Only one iteration of MAP adaptation on the model means were used for each of the speaker models.

The models were tested on half of the 2003 NIST SRE test set. The chosen subset of test data were exactly the same as those used in our segmental GMM system. According to the NIST 2003 Speaker Recognition Evaluation Plan, each test utterance was tested against the background model and 11 speaker models. Each speaker model score is normalized using the background model score on the same utterance.

We used T-norm (Auckenthaler et al. 2000) for score distribution scaling. During testing a set of example impostor models was used to calculate impostor log-likelihood scores for each test utterance, and from these scores a mean and variance were estimated. As we don't know which speaker is the true speaker and which is an impostor, by including 50 speakers in the cohort the mean and variance of the impostor cohort can be reliable (Auckenthaler et al. 2000). The true speaker, if included in the cohort which has 49 impostors, will not have a big effect on the mean and variance of the impostor cohort. Auckenthaler's work also concluded that a cohort size above 50 speakers leads to no significant improvement in speaker verification perfor-

mance. The 50 speakers from the impostor cohort were tested against each test utterance. The 50 impostor log-likelihood scores were treated as from a normal distribution and a mean and a standard deviation of the distribution were calculated. All verification scores for each utterance were normalized by subtracting this 'impostor mean' and dividing by the 'impostor variance'.

10.2 Experiment Results

The Speaker Verification performances of the three GMM systems are shown as DET curves in figure 10.1. Surprisingly the best result is achieved by the 'delta-only' system (sys_19d), with an EER between 9% and 10%. The static-only system, (sys_19s), achieves an EER of approximately 15%, and the system which employs both static and delta parameters has a performance better than the static-only system but worse than the delta-only system, with an EER around 12%. This pattern is clearly different from other published results, which typically show that the 'statics-plus-deltas' system works better than the 'static-only' system, and that both systems work better than the 'delta-only' system. We tried many different system settings, but the ranking of the results remained the same. The 'delta-only' system always gave the best performance. It's significant that the systems which perform poorly based on 'delta-only' parameter set are typically dealing with clean data (Soong and Rosenberg 1988; Liu, He, and Palm 1996).

A remaining difference between our system and others which perform well for TI-SV on Switchboard and achieve best results using 'statics-plus-deltas' parameters, is that our system does not include any provision for noise ro-

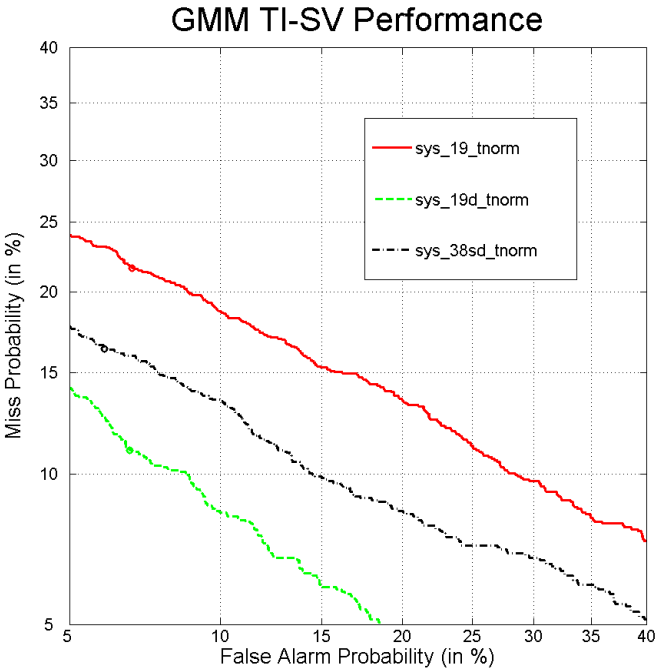


Figure 10.1: *TI-SV results using GMMs.*
red curve: static-only system with T-norm
black curve: static-plus-delta system with T-norm
green curve: delta-only system with T-norm

bustness other than CMS. Other systems use different techniques for noise robustness, for example, the voice activation detector (Appiah, Sasikath, Makrickaite, and Gusaite 2005), RASTA filtering (Hermansky and Morgan 1994), feature warping (Pelecanos and Sridharan 2001), handset score normalization (Reynolds et al. 2000), Biologically-Inspired Auditory Features (National Institute of Standards and Technology 2003), and so on. This led us to speculate that the superior performance of our 'delta-only' system was due to the robustness of the 'delta' parameters to noise which is obviously important for Switchboard. We then added RASTA filtering technique into the GMM systems.

10.3 RASTA filtering

One of the most popular approaches to noise robustness is RASTA filtering (Hermansky and Morgan 1994). RASTA filtering is motivated by the observation that human hearing is relatively insensitive to a slow change in the frequency characteristics of the communication environment and thus steady background noise does not severely impair human perception of speech. RASTA filtering uses a spectral estimate in which the time varying signal in each frequency channel is band-pass filtered in time. Hence RASTA filtering suppresses components of the time-varying signal in each channel which change either too quickly or too slowly.

A typical implementation of RASTA filtering begins with the transformation of the speech signal into a regular sequence of short-term critical-band log spectra. Next, for each spectral channel the temporal derivative is calculated

using a regression line through five consecutive time values. The sequence of spectra is then recomputed by integrating these derivatives through time in each spectral channel. This whole process is equivalent to first-order IIR filtering of each frequency channel time series.

RASTA should help the system deal with noises in the Switchboard data, as recordings in Switchboard are of telephone conversation speech distorted by the telephone-communication channel, slowly changing background noises and other noises.

It is common practice to apply RASTA filtering to the sequence of MFCC vectors, rather than log spectral vectors, and this was done in the current experiments (Hermansky and Morgan 1994; Milner 2002; Burget et al. 2007). The new static features, which were estimated using these dynamic features, will be less sensitive to both very slow variations and very fast frame-to-frame variations in the short-term cepstrum.

10.3.1 Experiment results

The results using RASTA filtering are shown in figure 10.2. This system in these experiments used CMS and variance normalization over each 3 second speech segment, an energy-based speech-noise detector and T-norm. The energy component, MFC_0, was discarded as a form of energy normalization. In this experiment the speech-noise detector had a threshold value of 16 (MFCC energy value) and discarded about 7% silence/noise data. After RASTA filtering, the new MFCC static features are different from the original features, as they were reconstructed using the delta features. Because

the "static" parameters are a function of the delta parameters only, it is correct to think of these "static" parameters, after RASTA filtering, are delta parameters. But for convenience of use I still call them "static" features.

The results show a substantial improvement in the performance of the 'static-only' system, sys_19, with a new EER of 10%. The performance of the 'delta-only' system is slightly worse, having an EER just above 10%. The 'statics-plus-deltas' system, sys_38sd, now gives the best performance, with an EER around 8%.

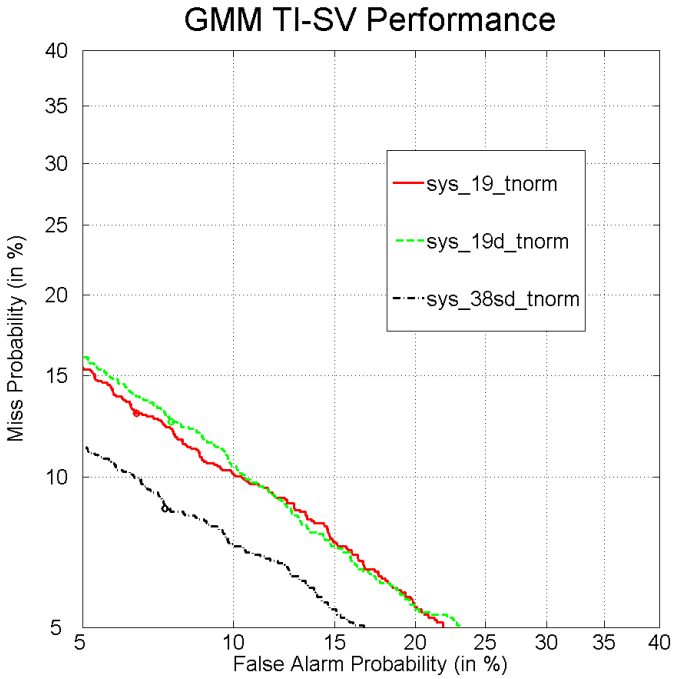


Figure 10.2: *TI-SV results using GMMs.*
red curve: static-only system with T-norm, and RASTA
black curve: static-plus-delta system with T-norm, and RASTA
green curve: delta-only system with T-norm, and RASTA

The shift of system performances, and the significant improvement in the speaker verification performance of the static-only system and the statics-

plus-deltas system, supports the hypothesis that the superior performance of the 'delta-only' system prior to RASTA filtering could be due to the noise-robustness of the deltas.

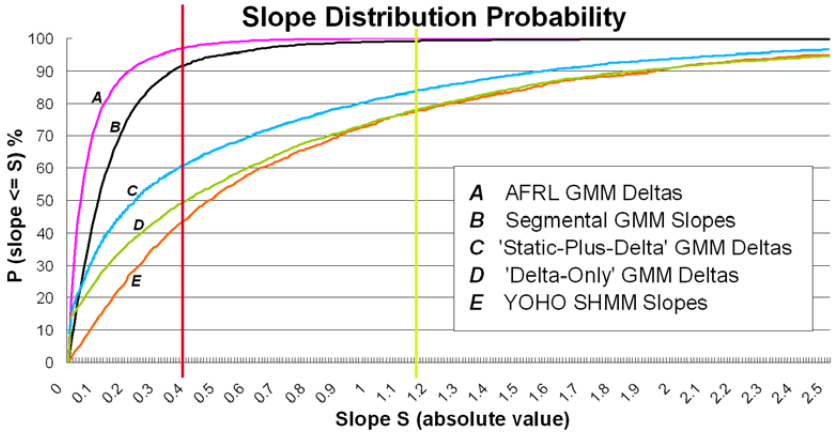
An interesting question is whether the deltas in the delta-only system and the statics-plus-delta system (with RASTA applied to them) contain dynamic information. In the process of RASTA filtering, the use of the delta features to re-construct the static features has a smoothing effect over the feature vectors. The components of the signal in each channel which change too quickly or too slowly will be restrained by the filtering. However, calculated over several consecutive frames (50ms in this case), the delta parameters should be able to capture some dynamic information within the 50ms range, after discarding unwanted fluctuation. We did some analysis to compare these deltas with our previous segmental model trajectories.

10.3.2 Further analysis of the 'delta' parameters

We calculated the cumulative distribution of the delta parameters in the background models of the delta-only system and the statics-plus-deltas system, with RASTA filtering applied to them. We also calculated the cumulative distribution of the delta values in the three systems from Figure 9.3 (the segmental GMM used for TI-SV on Switchboard, the phone-level SHMM used for TD-SV on YOHO, and the AFRL conventional GMM used for TI-SV on Switchboard). Figure 10.3(a) shows the cumulative distributions of the absolute delta values for the five systems. The red line and the yellow line indicate where the cumulative slope are 0.38 and 1.15 respectively. Figure

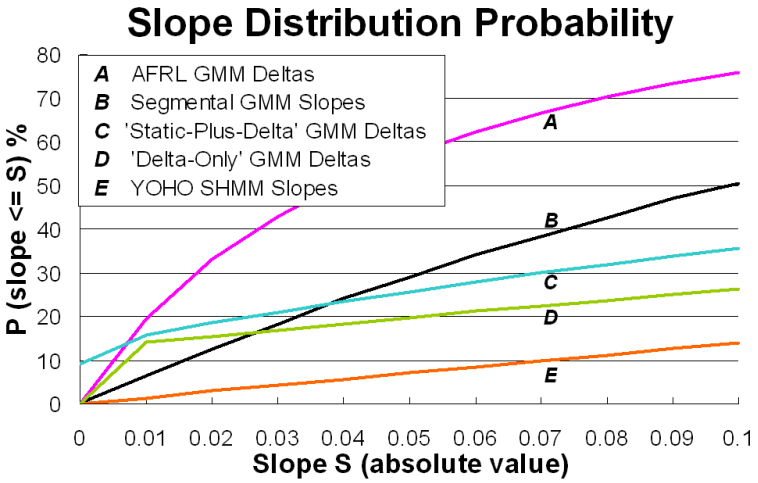
10.3(b) zooms in to show the probability of the delta value up to 0.1.

From the figures we can see that the distribution of the slopes/deltas are similar for the AFRL GMM system (curve A) and for our segmental GMM system (curve B). Both systems don't seem to model much dynamics. Compared to the AFRL GMM system and our segmental GMM system, the delta parameters in the conventional GMM systems we built (with RASTA applied to them) managed to capture more dynamic information. The UBM of the delta-only system (curve D) has bigger delta values than the UBM of the statics-plus-deltas system (curve C), although they started with the same initial deltas before training. It could be because that the delta-only system has half of the vector dimension than the statics-plus-deltas system, which leads to a wider range of distribution as there are less constraint during training from all other channels. Some experiments of fusing the two systems are presented in Chapter 11. The segment trajectory slopes of the SHMMs on YOHO (curve E) have much bigger values compared to the other four systems. The slopes are more diversely distributed and a greater percentage of slopes were to model the speech dynamics.



a. Delta/slope S in the range [0, 2.5])

The cumulative slope is 0.38 at the red line and 1.15 at the yellow line.



b. Delta/slope S in the range [0, 0.1]

Figure 10.3: Comparison of delta/slope distribution(*the delta-only and static-plus delta systems are after RASTA).*

Roughly 35% of the YOHO segments are distributed in the slope range of $[0.38, 1.15]$. As analyzed in Chapter 9, these segments contributed substaintially more to the speaker verification performance. Nearly 30% of the deltas in the delta-only GMM system and nearly 25% of the deltas in the static-plus-delta system are in the range of $[0.38, 1.15]$. This shows that apart

from being noise robust, the deltas in both GMM systems are modelling some important speech dynamics. As RASTA filtering has been applied to the two systems, it suggests that the unsupervised training on clean data is able to model some dynamics.

Less than 10% of segment slopes in the segmental GMM system and less than 5% of deltas in the AFRL GMM system are distributed in this range. For our segmental GMM system, future experiments using RASTA filtering should help improve the SV performance by removing unwanted noise and hopefully by modelling some of the important speech dynamics. For the AFRL GMM system, it is difficult to comment as we don't know exactly what the system settings are. However, although the AFRL GMM system works very well, if it can be trained to have more non-zero deltas especially the deltas which are in the range of $[0.38, 1.15]$, it is very possible that its performance could be further improved.

The role of the delta features appears to be different for different systems. In figure 10.3(b) we can see that although the static-plus-delta system contains more non-zero deltas than the AFRL GMM and segmental GMM systems, around 15% of its deltas are equal to or smaller than 0.01, and nearly 10% of its deltas are zeroes. The local optimality of the EM algorithm is a possible reason that these systems end up with different delta values after training. Alternatively, this may be due to other differences between these systems, such as the use of speech/noise detector or CMS. Whatever the explanation, the results show that the role of delta parameters in GMM-based speaker verification systems is more complex than simply "modelling dynamics".

10.4 Summary

We build conventional GMM systems each of which has a different set of feature vectors. Our results show, surprisingly, that prior to RASTA filtering systems based on 'delta' parameters alone outperform corresponding systems based on 'static-only' or 'static-plus-delta' parameters. However, after RASTA filtering the ordering is reversed, with the 'static-plus-delta' system performing best and the 'delta-only' system performing worse. This result suggests that the good performance of the 'delta-only' system may be due to their tolerance to noise (on which RASTA filtering relies) rather than their ability to capture speech dynamics.

Our experiments and analysis show that the role of the delta features varies. In the simplest case of TD-SV, they do model speech dynamics and the dynamic regions do contribute (more than the static regions) to SV performance.

In TI-SV on Switchboard, the story is more complex. Unsupervised ML training focuses on modelling the stationary regions and hence results in deltas close to zero. In recognition this has the fortuitous effect of focusing the classification process onto the stationary regions and this seems to improve performance. In particular, an acoustic vector which matches the mean but occurs in a dynamic region will get a small probability but the same vector in a static region will get a high probability. In general this seems to improve performance. The situation is further complicated by noise. For a corpus like Switchboard, if there is no noise compensation (such as RASTA), the "delta-only" system works better than the "static-only" system. This suggests

that the delta parameters are more noise tolerant than the statics. After RASTA filtering, the specific noise to which the delta parameters are robust is removed. This removes the "noise robustness" benefit of the deltas and allows the statics to perform well. Also, it appears that the deltas in the "delta-only" and "static-plus-delta" systems (with RASTA applied to them) are modelling dynamics.

Two further experiments could be done. One is TI-SV on YOHO, to see what happens if there is no noise in TI-SV. The other is segmental GMM on RASTA-filtered Switchboard, to see how trajectory model performs in the situation where the noise which affects the statics is removed. Supervised training can also be used for the segmental GMM system, to fix the segment slopes to be non-zero values.

Our analysis shows that the deltas are being used in different ways in different systems. It could be because different front-end processing, different system settings, or the local optimality of the EM algorithm. The role of delta parameters in GMM-based text-independent speaker verification systems is more complex than simply modelling dynamics.

CHAPTER 11

Fusion of the 'delta-only' and 'static-plus-delta' systems

According to the analysis of the delta-only and static-plus-delta GMM systems on Switchboard (with RASTA applied to them) described in the previous Chapter, the delta values are different in the delta-only and static-plus-deltas systems after ML training. The delta-only system, due to smaller dimensions in the feature vector, has more freedom to model speech dynamics during ML training. The static-plus-delta system however has a tendency to focus on stationary regions during ML training. Moreover, the delta-only system performs surprisingly well. These facts suggest that it may be advantageous to base the verification decision on a combination of the scores from the two systems.

Consequently we investigated a 'fused' score function of the following form:

$$S_f = \lambda * S_{s+d} + (1 - \lambda) * S_d, \tag{11.1}$$

where S_{s+d} is the score from the static-plus-delta system, S_d is the score from the delta-only system, and S_f is the fused score. The fusing parameter λ is a value in $[0, 1]$, which is determined empirically. When λ equals to 0, the fused system is actually the delta-only system; when λ equals to 1, the fused system equals the static-plus-the delta system.

11.1 Fusion results

The result of fusing the 'static-plus-deltas' and 'delta-only' scores, before T-norm is applied to either score, is shown in figure 11.1. The fused system works better than either the 'static-plus-delta' system or the 'delta-only' system, and the best performance was gained when the fusing parameter λ was set to 0.2. The effect of different values of λ is shown in figure 11.2.

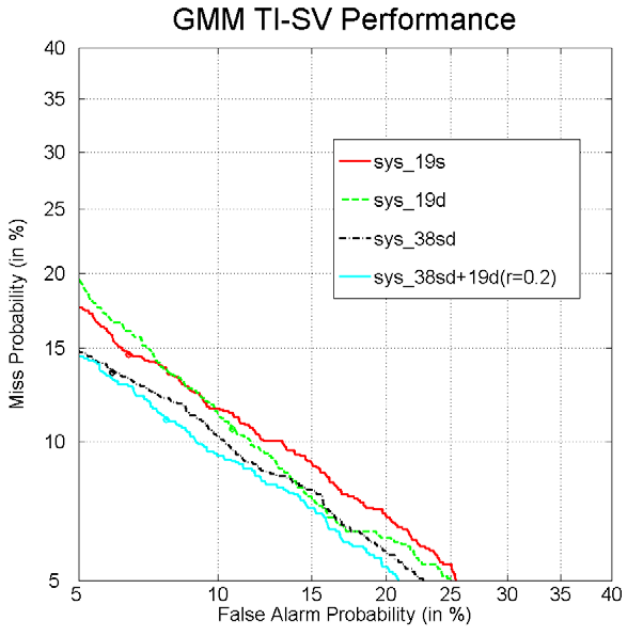


Figure 11.1: *Fusion of two systems without t-norm.*

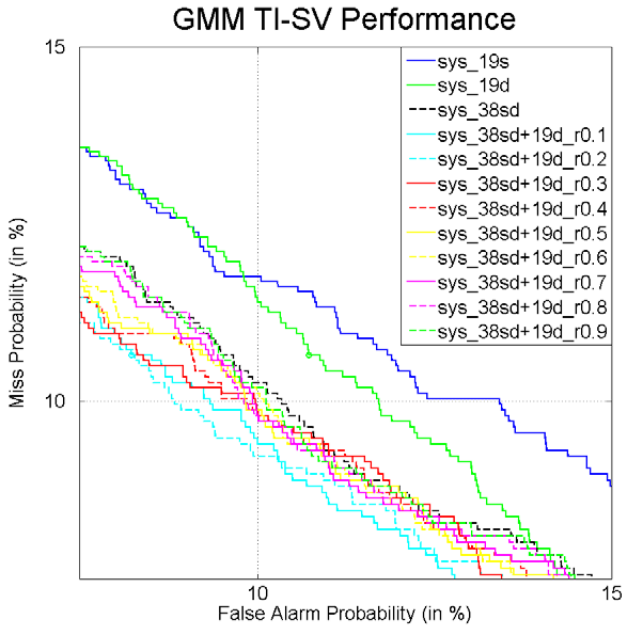


Figure 11.2: *Fusion of two systems without t-norm (effect of* λ*,* $\lambda = 0.1, 0.2, ...0.9$*).*

However, the performance obtained by fusing the two systems without T-norm is inferior to the performance of the 'statics-plus-delta' system on its own with T-norm (figure 10.2). Figure 11.3 shows the result of fusing the two systems after T-norm has been applied to both of them. The effect of different λ is shown in figure 11.4. In this case fusion appears to offer no advantage. This could be because that after T-norm, the score distributions are normalized so that the correlation between the two score distributions increases.

The fusion results indicate that, at least after T-norm, the scores of the delta-only and static-plus-delta systems are correlated.

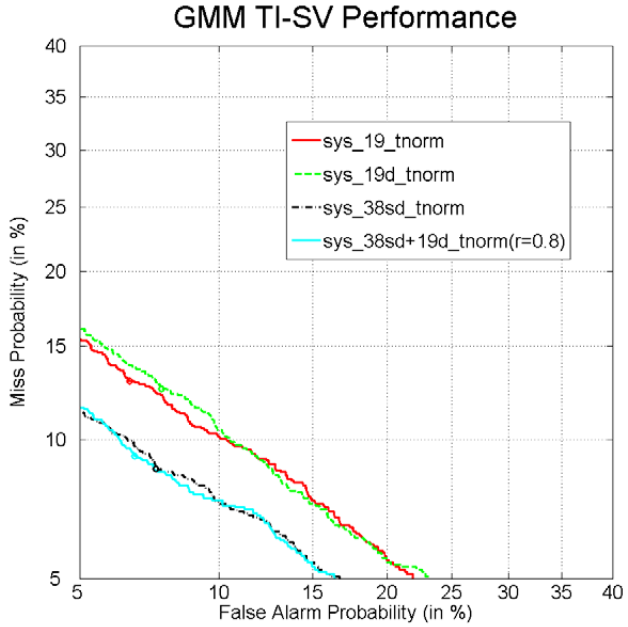


Figure 11.3: *Fusion of two systems after t-norm.*

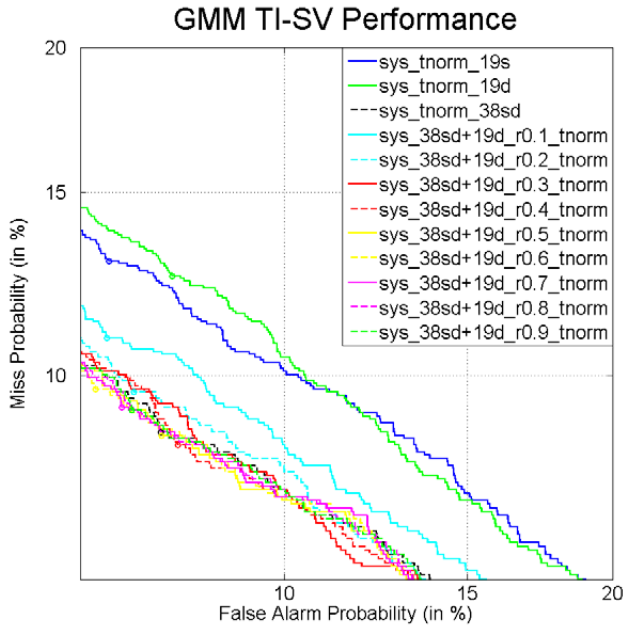


Figure 11.4: *Fusion of two systems after t-norm (effect of* λ*,* $\lambda = 0.1, 0.2, ...0.9$*).*

11.2 Correlation of the 'delta-only' and 'static-plus-delta' scores

To investigate the correlation of the 'delta-only' and 'static-plus-delta' scores, we plotted the following figures. Figures 11.5(a) and (b) show scatter plots of the scores for the 'delta-only' system against the corresponding scores for the 'statics-plus-delta' system. Figures 11.5 and 11.6 are for scores before and after application of T-norm, respectively. The red 'x's are the true speaker scores. The blue 'o's are the impostor scores.

In both cases the scores from the 'delta-only' system and the 'static-plus-delta' system are clearly correlated, as one would predict from the small effects gained by fusing the two systems.

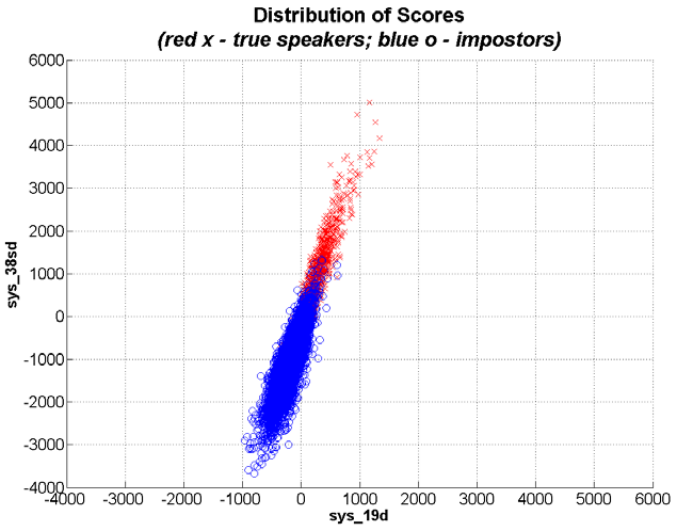


Figure 11.5: *Score distributions of the "delta-only" system and the "static-plus-delta" system (with RASTA applied to them, without T-norm).*

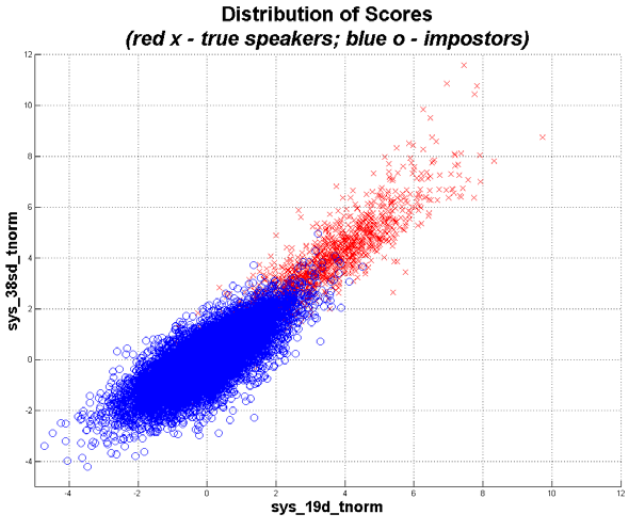


Figure 11.6: *Score distributions of the "delta-only" system and the "static-plus-delta" system (with RASTA, and T-norm applied to them).*

11.3 Summary

The different deltas in the delta-only and static-plus-delta systems led us to attempt to fuse the two systems. Fusion of the two systems without T-norm being applied to either score works better than either system. Fusion of the two systems with T-norm doesn't improve the performance. We have shown that the scores produced by the delta-only and static-plus-delta systems are correlated, and that there is little to be gained by fusing the two systems.

CHAPTER 12

Conclusion and Discussion

This thesis has introduced the trajectory-based segmental hidden Markov model as a new speaker model for speaker verification. We applied the linear trajectory segmental models to text-dependent speaker verification on YOHO and text-independent speaker verification on Switchboard. During the process we studied the behavior of the model to understand the role of dynamic features in speaker verification systems. The following sections will present the results, analyses and draw conclusion on the work.

Conventional stochastic modeling methods have the inappropriate assumptions of independence and piece-wise stationarity of the sequences of acoustic vectors which constitute a speech pattern. Segmental HMMs try to overcome this problem by associating states with sequences of acoustic vectors, rather than individual vectors. In this way it is possible to model correlations between acoustic vectors in the sequence. The segmental HMM used in this thesis has an incorporated linear trajectory which describes how the features change over time in the segment. As we know, during the pro-

duction of speech, the formant transition comes from changes in the shape of the vocal tract, which varies from speaker to speaker. Some previous studies show that the dynamic spectral regions are different for different speakers. We have also cited evidence which indicates that in some cases speech recognition is not influenced by the details of these dynamic regions, and hence there is scope for individual differences. With improved modeling of speech dynamics and duration, the segmental HMM should have advantages for speaker verification. For model parameter estimation, the segmental Viterbi decoder, which uses an EM algorithm, provides an iterative maximum likelihood estimation technique.

We applied the segmental HMM in text-dependent speaker verification on the YOHO database and the segmental GMM in text-independent speaker verification on Switchboard. The segmental models managed to improve performance in TD-SV but failed to prove any benefit for the TI-SV system. To understand why segmental models work differently in these two systems further investigation has been carried out, in which the segmental models also act as a tool to help us explore the role of dynamic features in text-independent and text-dependent speaker verification.

12.1 TD-SV and TI-SV using segmental HMM

The results of the text-dependent speaker verification experiments on YOHO show a 25% reduction in the number of false rejections and a 44% reduction

in the number of false acceptances by using the segmental HMM system, relative to the conventional HMM-based system. As we expected, the better modelling of speech dynamics and duration of the segmental HMM system helps capture some individual characteristics and hence improve the speaker verification performance. A flaw with the TD-SV experiment on YOHO is that the two systems we compare have different numbers of parameters, hence the SHMM has an unfair advantage. But as the SHMM system and HMM system have different model strategies, it is difficult to build an HMM system which has exactly the same number of parameters as the SHMM system. We did some experiments to reduce the number of parameters of the SHMM system instead, which gives us some references of how the two system perform under the same condition of the number of parameters. Our conclusion is that the SV performances of the HMM and SHMM systems can not be simply judged by the number of parameters in the two systems. Further analysis which compares the segment-level speaker verification scores unveils that the segments with non-zero slopes do contribute more to the speaker verification performance than the segments with zero slopes. To be particular, the segments which produce both significantly higher true speaker scores and significantly lower imposter scores are in the slope range of $[0.38, 1.15]$, which makes it a very important dynamic range for speaker verification.

For text-independent speaker verification, a segmental GMM based system was constructed using the Switchboard speech corpus (trained on NIST 2002/3 and tested on NIST 2003). We performed two main sets of experiments, one of which was to test the effect of different segment slopes, the

other was to test the effect of different segment lengths. Disappointingly all the performances from these two experiments are very close and the equal error rates are all approximately 14%. It seems that we can't get any benefit from using the segmental GMM with non-zero trajectory slopes.

For segmental modelling the computational load is significantly greater than that associated with a conventional GMM. An important issue is to reduce this computational cost. The 'SEGVit' software toolkit has been modified so that model training can be conducted in parallel on a cluster of computers. We also investigated a technique introduced by Auckenthaler, which exploits the link between a component of the UBM and the corresponding component of each of the SDMs. Once the optimal sequence of components has been computed for the UBM, we used exactly the same sequence for each of the SDMs. Because in this case we do not need to recalculate the duration for each of the SDMs, it saves us a large amount of computer running time.

The Switchboard results contrast with the previous YOHO results, which show obvious benefits from using segmental HMMs. Further analyses were carried out to study this issue, which shed a light on the role of dynamic features in TD- and TI-SV systems, including conventional GMM-based systems.

12.2 The Role of Dynamic Features

We visualized the individual segments of the segmental GMMs by applying an inverse Discrete Cosine Transform to the MFCC vectors in the UBM. It

turned out that most of the visualized segment models contain quite flat trajectories, which suggests that they do not carry much dynamic information. This led us to investigate the distribution of slope values in our segmental GMMs. In fact, nearly thirty percent of the segment trajectories in the UBM are very close to zero, although the initial values of these trajectories before training were not nearly zero. It seems that after training, the distribution of the trajectory values moved towards zero.

We then did experiments to see the effects of different sets of MFCCs on the trained UBM. The results indicate that as the number of MFCCs increases, or , as the number of segments decreases, the percentage of non-zero slopes in the trained UBM decreases. This suggests that the lack of non-zero slopes is due to the maximum likelihood training algorithm giving priority to modelling stationary regions. Looking into the distribution of slope values in a conventional GMM system from AFRL, which was also trained on Switchboard, confirmed this theory. Most of the delta parameters in the conventional GMM background model are nearly zero. Consequently only few delta parameters in either system model the dynamics in the range of $[0.38, 1.15]$. The delta features seem to focus the system onto the stationary region of the speech, rather than to represent the dynamic regions.

Compared to GMMs and segmental GMMs trained on Switchboard, segmental HMMs trained on YOHO have much bigger slope values. We investigated the correlations between the SHMM trajectory slopes and the SV scores. It is shown that most of the segments in the nonzero-slope models produce bigger segment-level true speaker scores and smaller segment-level impostor scores compare to the zero-slope models, hence increase the

speaker verification performance. This is most prominent in the slope range of $[0.38, 1.15]$, which has about 35% of the total segments. Thus the SHMMs in a TD-SV system manage to contain speech dynamic information and these dynamic regions do contribute to speaker verification accuracy more than the stationary segments.

The results indicate that in a TD-SV system based on phone-level models with supervised training, dynamic structure can be exploited to improve performance. This is because the requirement to model explicit phone-level units encourages the models to take account of non-stationary regions. However, for a TI-SV system based on segmental GMMs with unsupervised training there is no such constraint. When the database is a simple one such as YOHO, a GMM system with enough number of states is able to model dynamics. However, for a noisy and complicated database such as Switchboard, it is difficult to model dynamics. Without a supervised training, the maximum likelihood training focuses on vectors which are close to the state mean and in stationary regions, as they give higher probabilities than the vectors which are close to the state mean but in dynamic regions. Discriminative training is a possible way to improve the correctness of the model as it uses a different objective function. The fact that non-stationary regions in the TD-SV system contribute more to discrimination indicates that they might be preferred in a discriminative training approach.

It is difficult to compare our segmental GMM system directly to the AFRL system, which is a conventional GMM system, because we do not fully understand the detailed training decisions which were made in the AFRL system. To try to understand better the role of delta features in a conventional GMM-

based SV system, and to understand how the dynamic features are affected by other components of the system, we built our own state-of-the-art GMM system.

12.3 The role of Delta Features in a GMM TI-SV system

First, we built the conventional GMM TI-SV systems each of which contain different feature sets: static parameters only, delta parameters only, and static plus delta parameters. We use the same Switchboard material which was used in our segmental GMM system (NIST 2002 and NIST 2003) to train and test these systems. HTK was used to build and to test these TI-SV systems. Different from other published results, our results show that the best result is achieved by the 'delta-only' system, followed by the 'static-plus-delta' system. The 'static-only' system achieves the worst performance of the three. Results from other systems typically show that the 'static-plus-delta' system works better than the 'static-only' system, and that both systems work better than the 'delta-only' system. As our system does not include any provision for noise robustness other than CMS, we speculated that the superior performance of our 'delta-only' system could be due to the robustness of the 'delta' parameters to noise.

We then added RASTA filtering to our GMM systems. Our results show that after RASTA filtering the ordering is reversed, with the 'static-plus-delta' system performing best and the 'delta-only' system performing worse.

This result suggests that the good performance of the 'delta-only' system is mainly due to the tolerance of the delta features to noise (on which RASTA filtering relies). It also suggests that in a conventional GMM TI-SV system, the contribution of the dynamic features to the TI-SV performance may also be mainly due to their robustness to noise.

12.4 Summary

Comparison between and analysis on the five different systems (TD-SV using segmental HMM on YOHO, TI-SV using segmental GMM on Switchboard, AFRL TI-SV system on Switchboard, 'static-plus-delta' GMM TI-SV system with RASTA on Switchboard, and 'delta-only' GMM TI-SV system with RASTA on Switchboard) help us draw the following conclusions.

- The segmental HMMs can model speech dynamics in TD-SV. With supervised ML training, the dynamic regions in the speech patterns can be modelled thanks to the explicit phone-level modelling which forces the ML training to take account of the dynamics. The analysis show that most of the nonzero-slope trajectory segments generate higher true speaker scores and lower impostor scores, compared to the zero-slope trajectory segments. This is especially prominent for the slope range of $[0.38, 1.15]$. This confirms that the dynamic regions in speech patterns, especially the dynamics of the range $[0.38, 1.15]$, contain important speaker information; the linear trajectory segmental HMMs can be used to model speech dynamics; the implementation of the speech dynamics can help increase the speaker verification accuracy.

- In the TD-SV experiments on YOHO the SHMM system has more parameters than the HMM system. This is because we can not make a SHMM system have the same number of parameters as the HMM system. However, due to the different model strategies between the HMM and SHMM systems, the performances of the two system can not be judged simply by the different number of parameters. The HMM and SHMM systems are two totally different systems and each has its own ways of using their parameters. The different duration pdfs of the two systems, the maximum duration set in the SHMM systems, and maybe other unknown factors could all affect the SV performances.

- The segmental GMMs fail to model speech dynamics in TI-SV. The segment trajectories are very close to zero after training, with only few segments modelling dynamics in the range of $[0.38, 1.15]$. The same phenomenon happens in the AFRL GMM system, with the deltas being dragged towards zero after the training. In TI-SV, without any particular supervision, the ML training favours stationary regions as these regions give higher probabilities during training. The discriminative training is a possible direction to improve the model training to exploit the utilities of non-stationary regions and hence make it more suitable for the speaker verification purpose.

- Without RASTA filtering, the 'delta-only' GMM system works best. After RASTA filtering, the performance of the 'static-plus-delta' GMM system improves substantially and becomes best. The results suggest that the good performance of the 'delta-only' system before RASTA

is mainly due to the noise robustness of the delta parameters. Analysis also shows that the deltas in both systems are modelling speech dynamics.

- Noise removal techniques such as RASTA are very important to a speech corpus such as Switchboard, which contains noisy telephone conversations. Without using these techniques, even the linear trajectory segmental models struggle to model speech dynamics due to the compromises made by the noise. Experiments using segmental GMMs on Switchboard after RASTA filtering, or, experiments using segmental GMMs on YOHO (TI-SV), can be done to test the ability of segmental GMMs to model speech dynamics on clean data under unsupervised ML training.

- A GMM TI-SV system on YOHO manages to model the dynamics. YOHO has a limited vocabulary. Each word of the YOHO vocabulary has a sufficient number of high quality training samples. This helps the TI-SV systems to accurately model the dynamics, provided that there is enough states to model the data. Switchboard, on the contrary, has a large vocabulary and quite noisy data. The complications of the Switchboard data, together with the priority of the ML training to model the static regions, make it a more difficult task to model the speech dynamics.

- The role of the delta features varies in different TI-SV systems. The different functions of deltas may be due to the local optimality of the EM algorithm, the different system setting (e.g. the number of states) and

the front-end processing (e.g. RASTA, speech noise detector, CMS). This indicates that the role of delta parameters in GMM-based TI-SV systems is more complex than simply "modelling dynamics".

- Finally, we have shown that the scores produced by the 'delta-only' and 'static-plus-delta' systems are correlated, and that there is little to be gained by fusing the two systems.

APPENDIX A

Effects of reducing the computational load

Results of experiments to investigate the effect of using the UBM optimal state sequence when computing the SDM probabilities, and different values of λ_1 and λ_2.

The DET curves for the systems which use the optimal UBM state sequences when calculating SDM probabilities are shown with a solid line (this is the 'Auckenthaler method'). The DET curves for systems which apply Viterbi decoding separately to the UBM and SDMs are shown with a dashed line. This line is the same in all of the figures and corresponds to $\lambda_1 = 1$; $\lambda_2 = 0$.

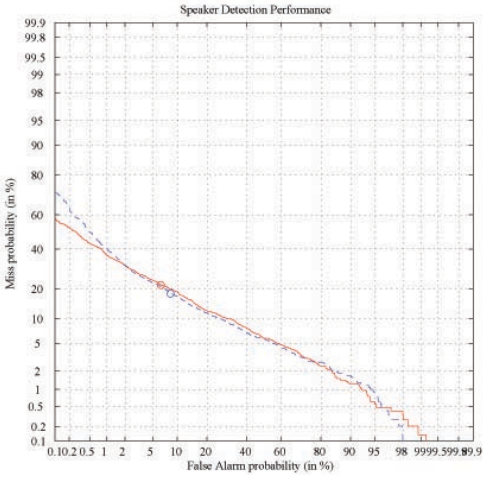


Figure A.1: $\lambda_1 = 1$; $\lambda_2 = 0$

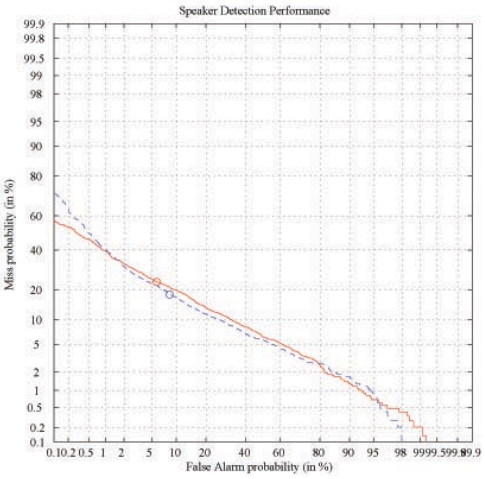


Figure A.3: $\lambda_1 = 1$; $\lambda_2 = 5$.

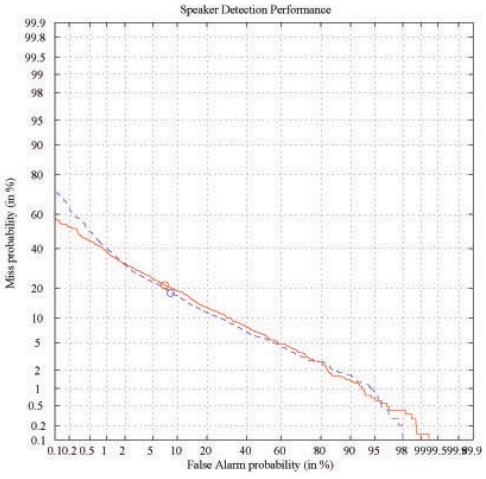


Figure A.2: $\lambda_1 = 1$; $\lambda_2 = 2$.

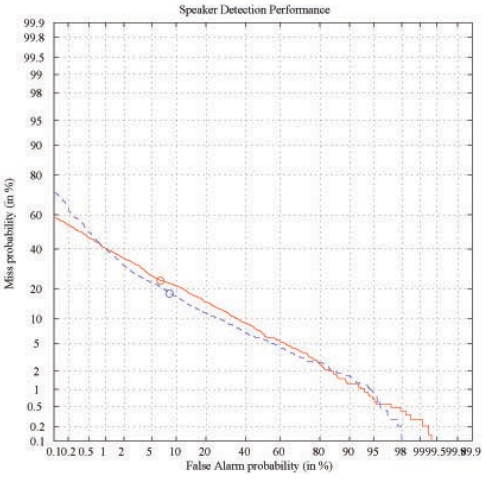


Figure A.4: $\lambda_1 = 1$; $\lambda_2 = 15$.

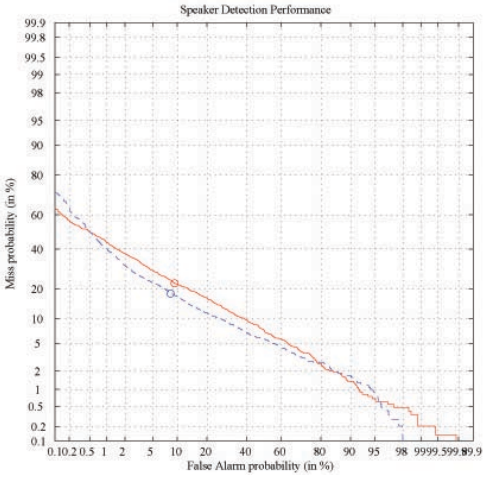


Figure A.5: $\lambda_1 = 1$; $\lambda_2 = 50$.

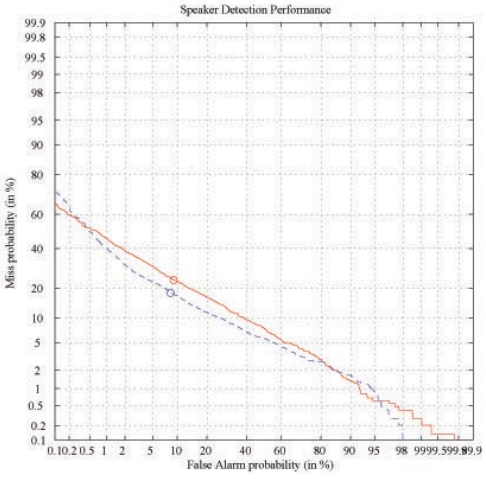


Figure A.6: $\lambda_1 = 1$; $\lambda_2 = 100$.

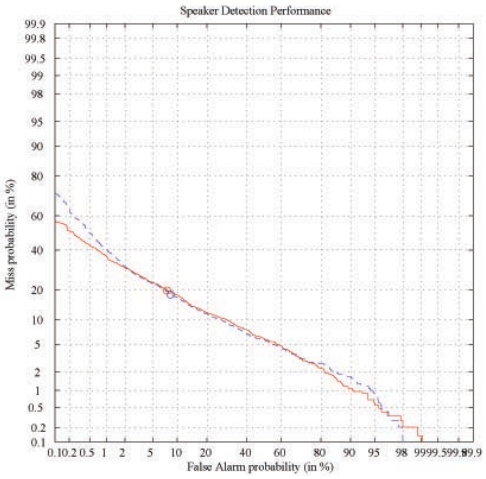


Figure A.7: $\lambda_1 = 1$; $\lambda_2 = -2$.

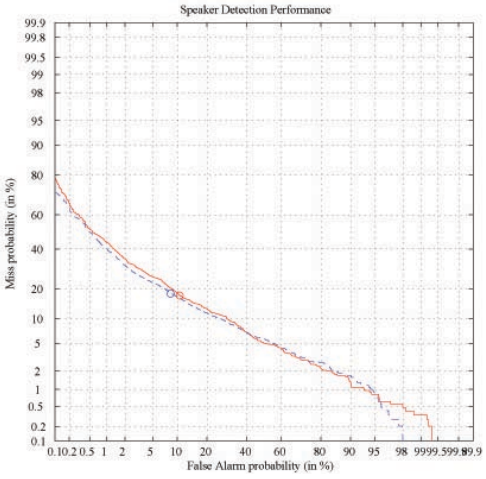


Figure A.8: $\lambda_1 = 1$; $\lambda_2 = -10$.

APPENDIX B

Results of applying λ_1 and λ_2

Results of experiments to investigate the effect of using the UBM optimal state sequence when computing the SDM probabilities, and different values of λ_1 and λ_2. Experiments are as in Appendix A, except that the same values of λ_1 and λ_2 are used in training and recognition.

The DET curves for the systems which use the optimal UBM state sequences when calculating SDM probabilities are shown with a solid green line (this is the 'Auckenthaler method'). The DET curves for systems which apply Viterbi decoding separately to the UBM and SDMs are shown with a dashed line ($\lambda_1 = 1$; $\lambda_2 = 0$). The DET curve for a conventional GMM system is shown with a solid, black line.

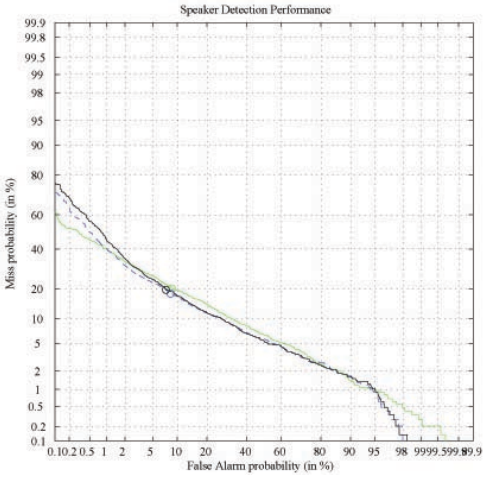


Figure B.1: $\lambda_1 = 1$; $\lambda_2 = 5$.

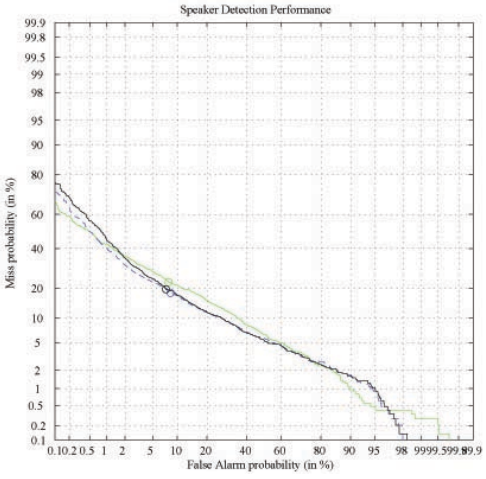


Figure B.2: $\lambda_1 = 1$; $\lambda_2 = 15$.

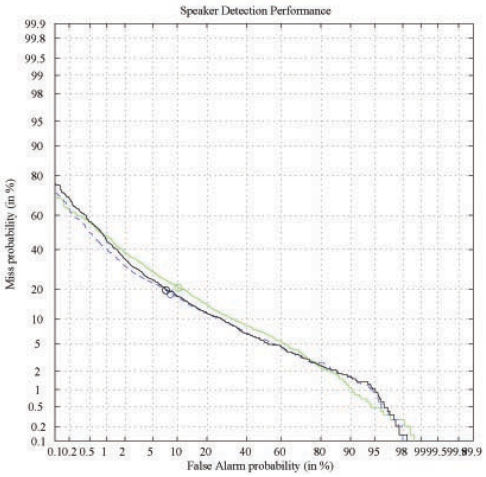
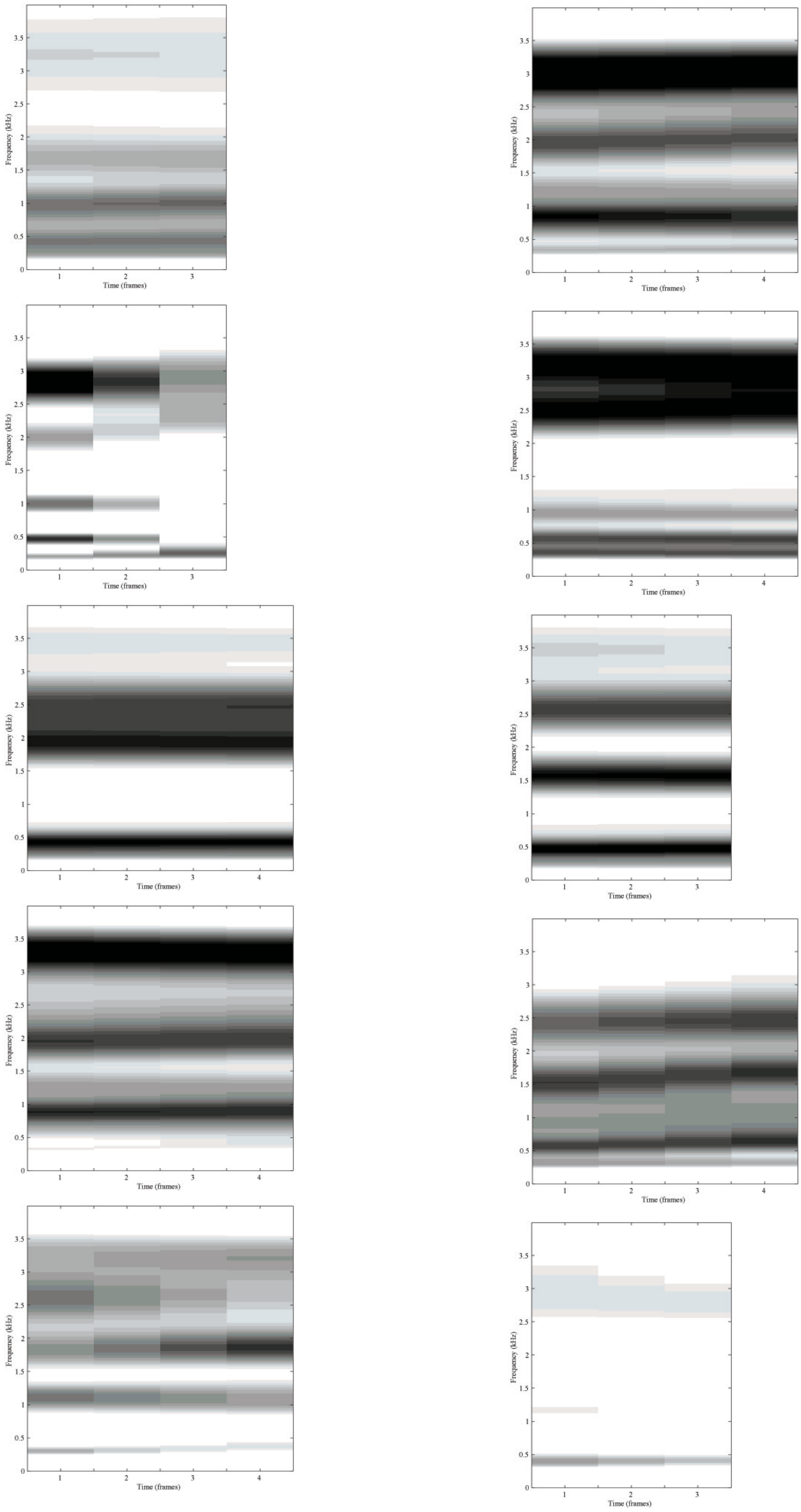


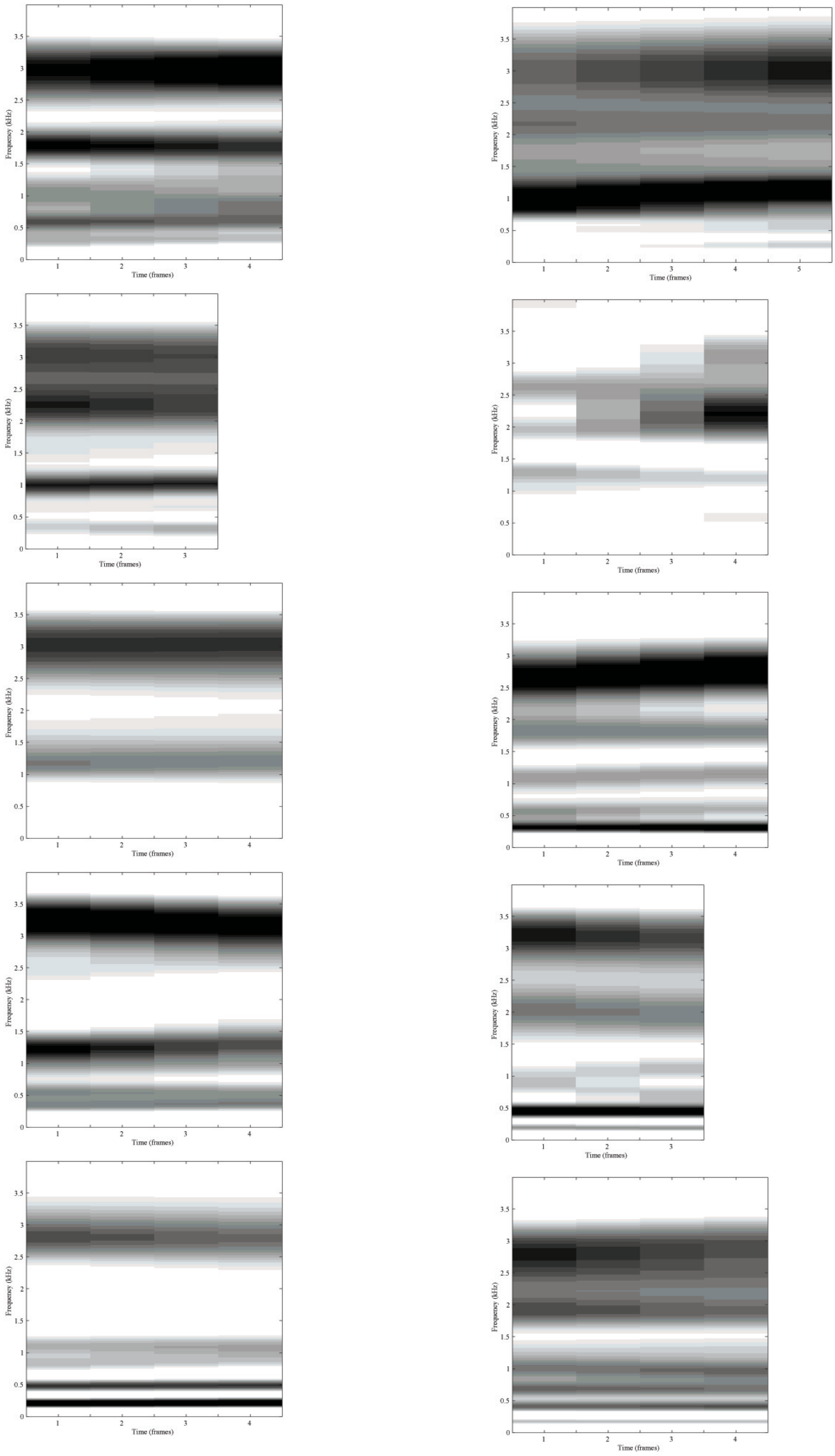
Figure B.3: $\lambda_1 = 1$; $\lambda_2 = 50$.

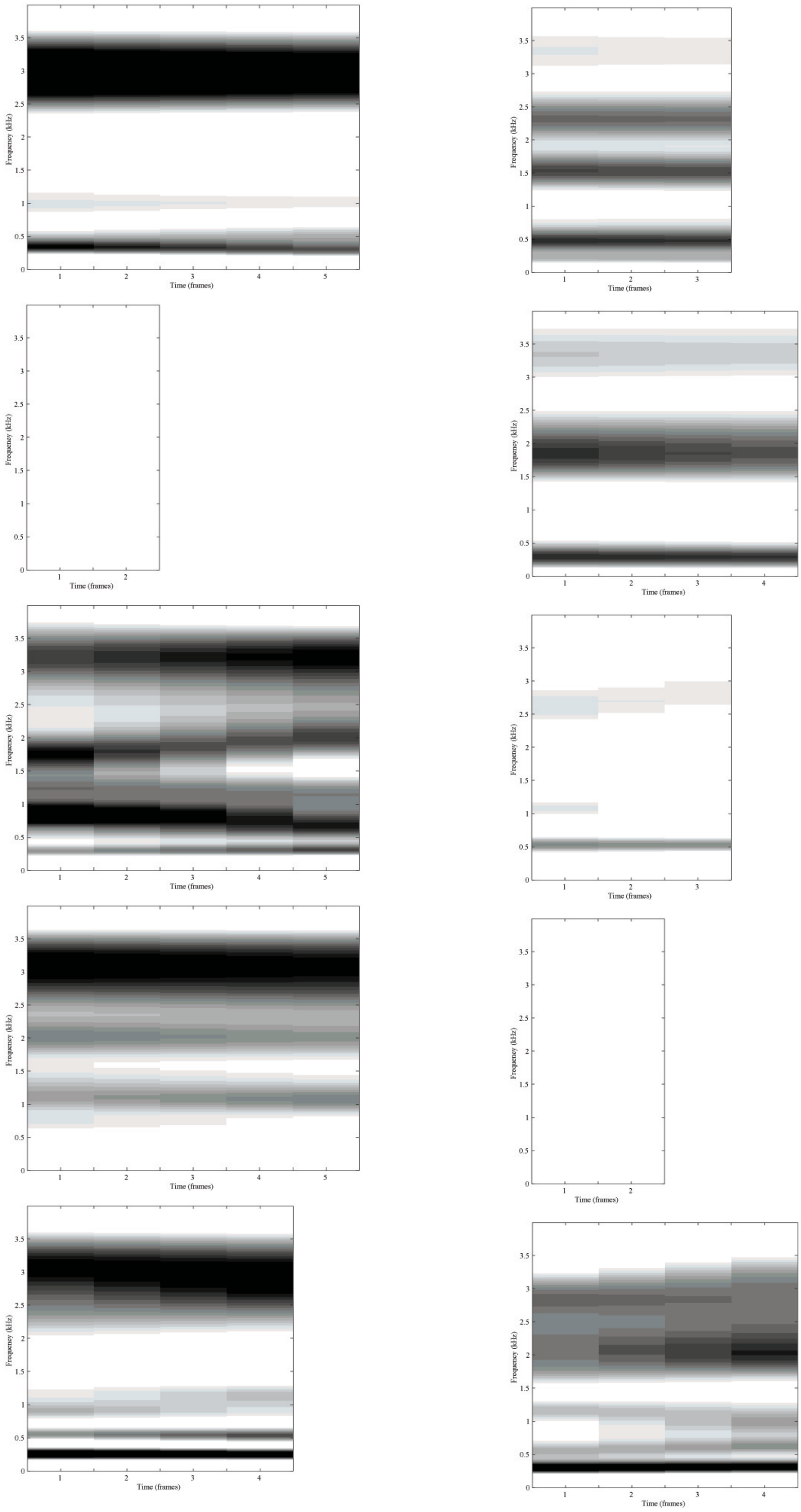
APPENDIX C

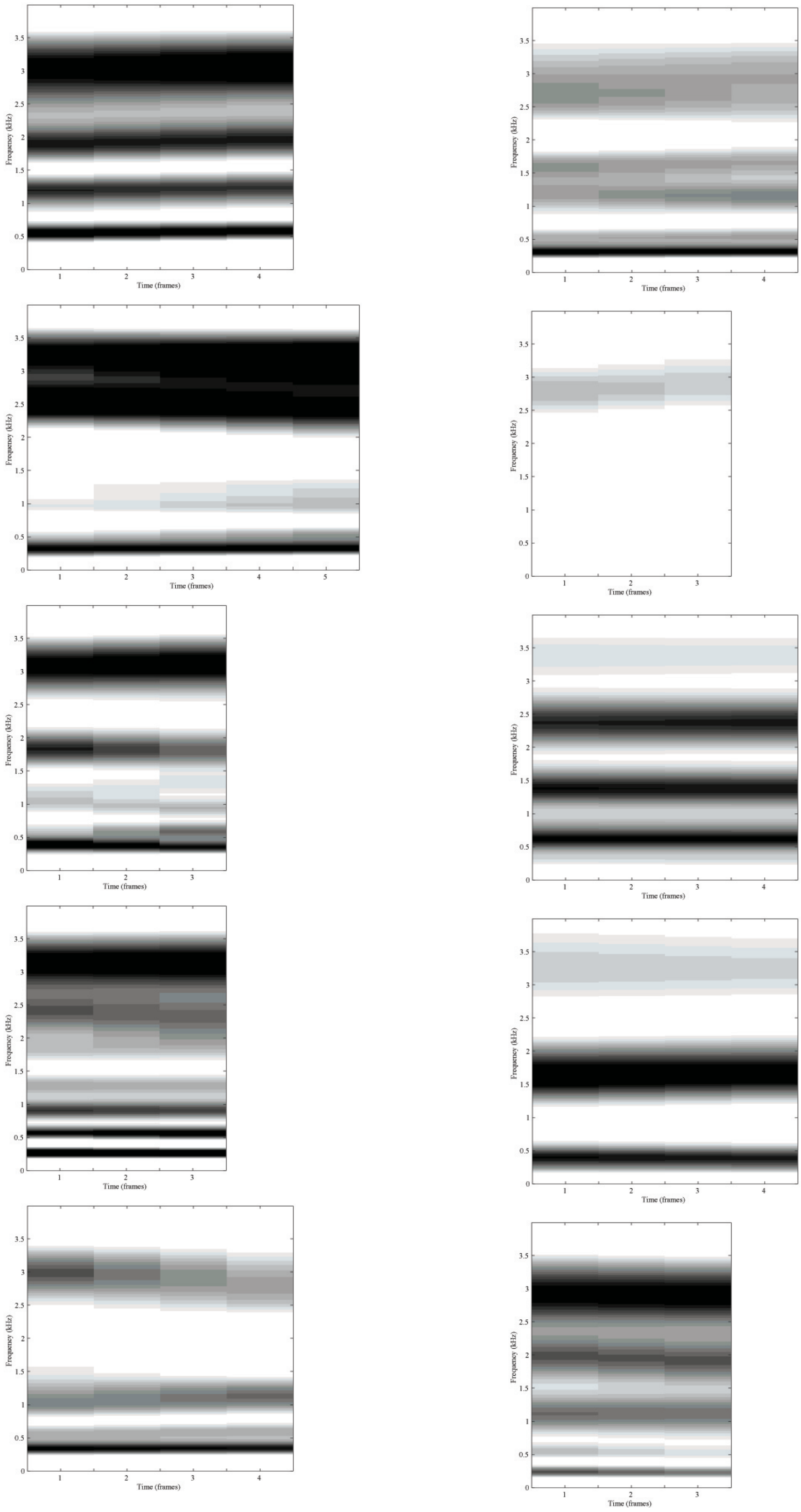
Visualization of the segmental GMMs

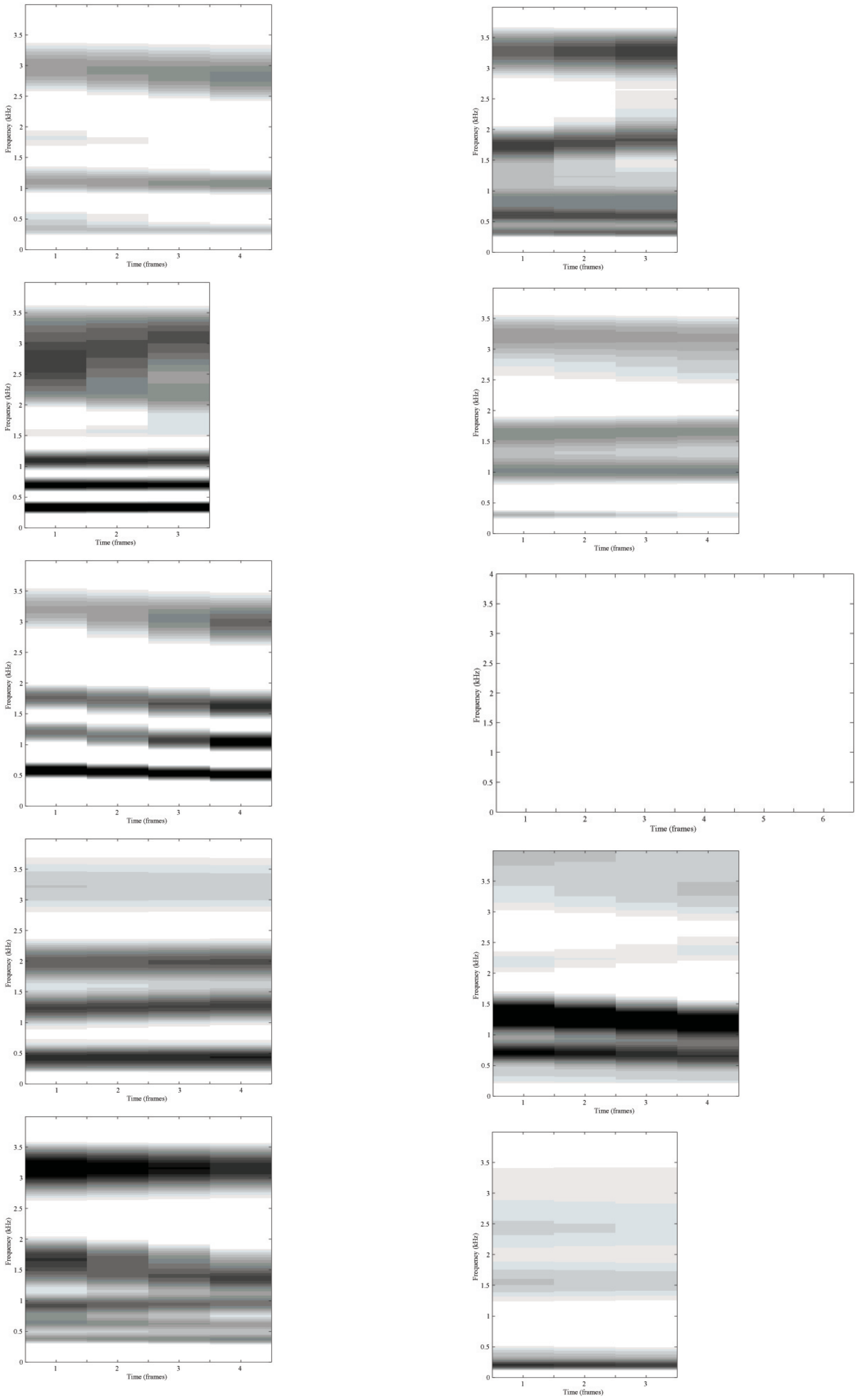
Spectrograms corresponding to 300 trained segments from the speaker dependent model for female speaker 5090

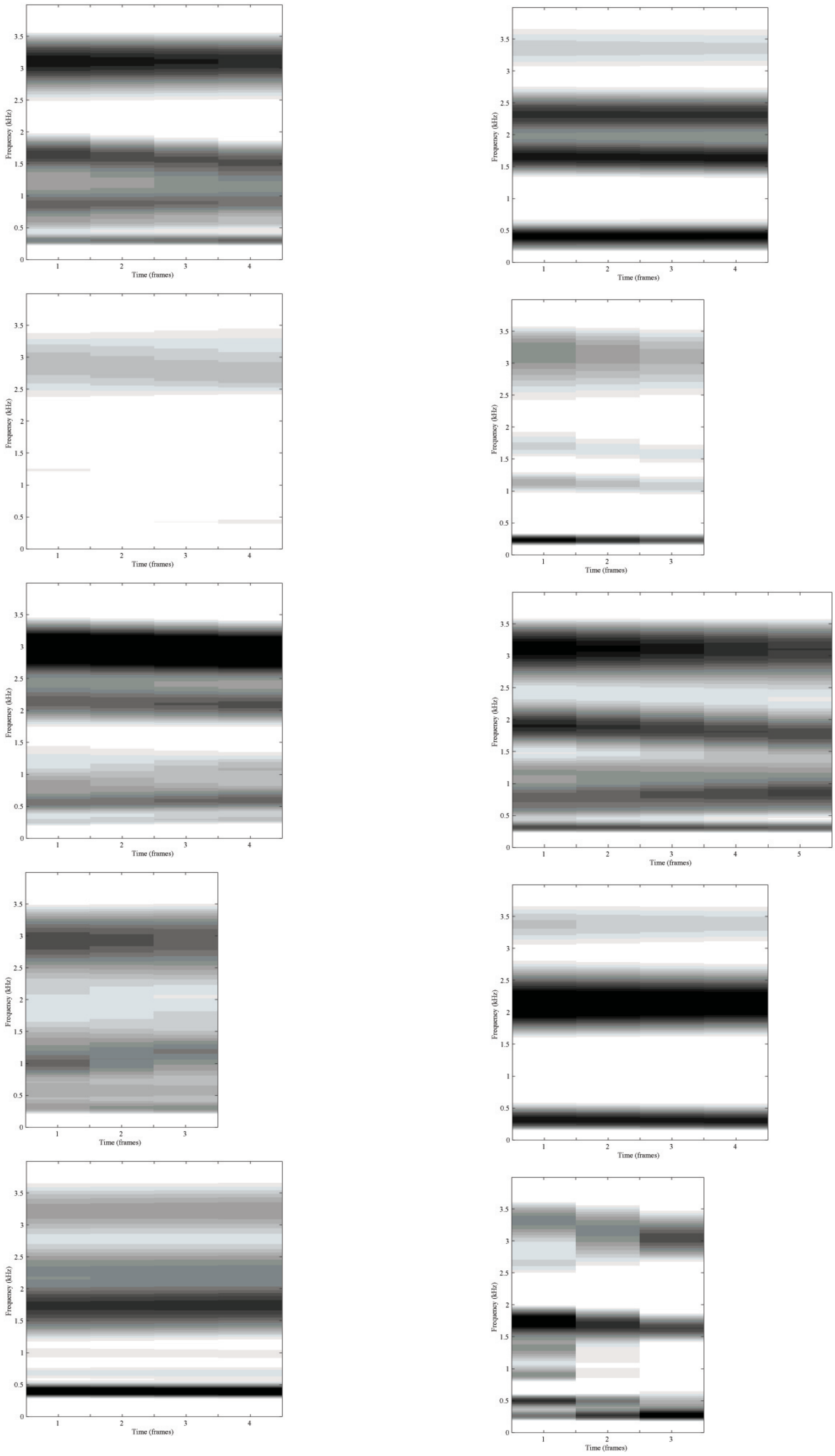

Frequency (kHz)
Time (frames)

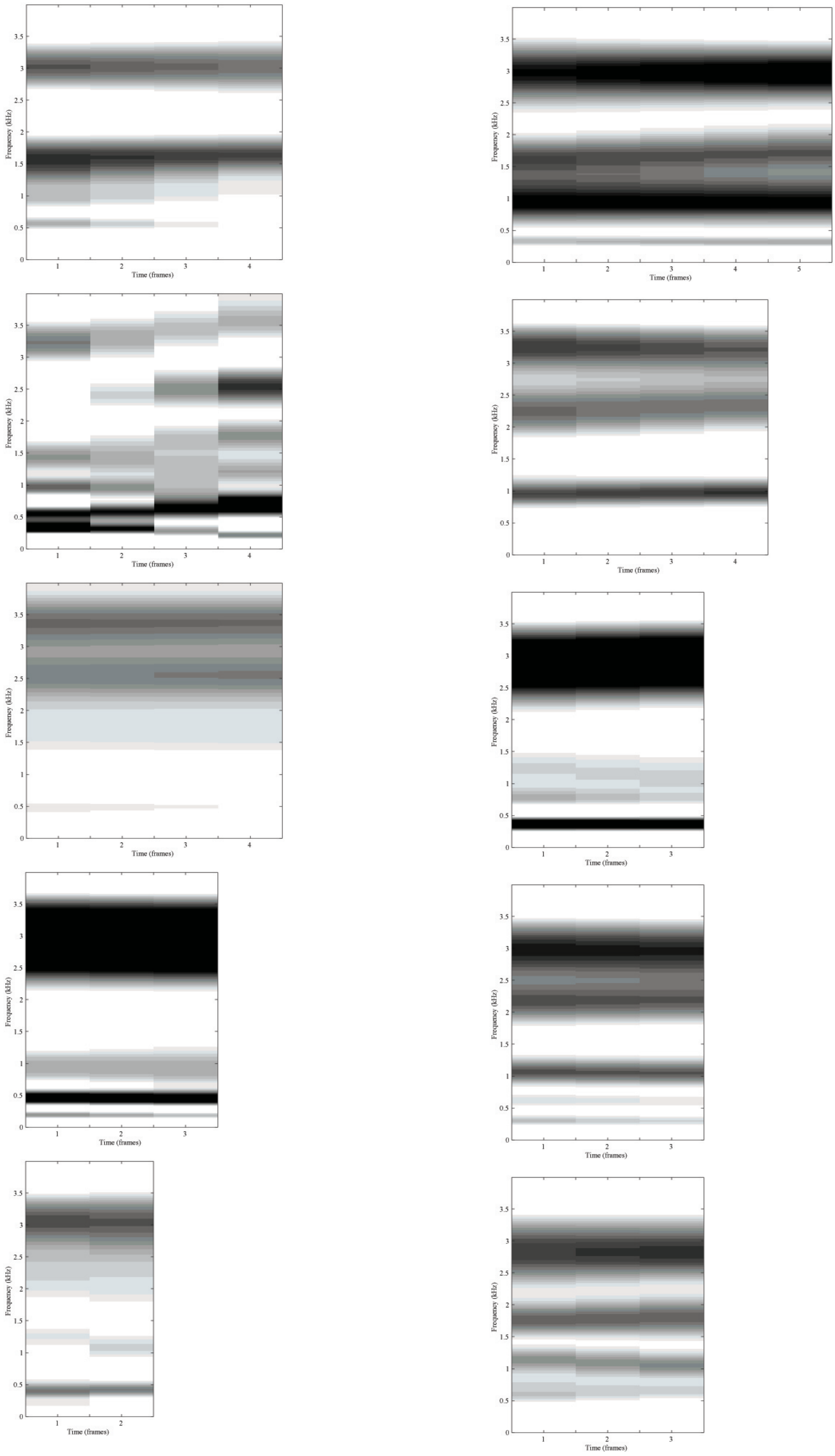

Frequency (kHz)
Time (frames)

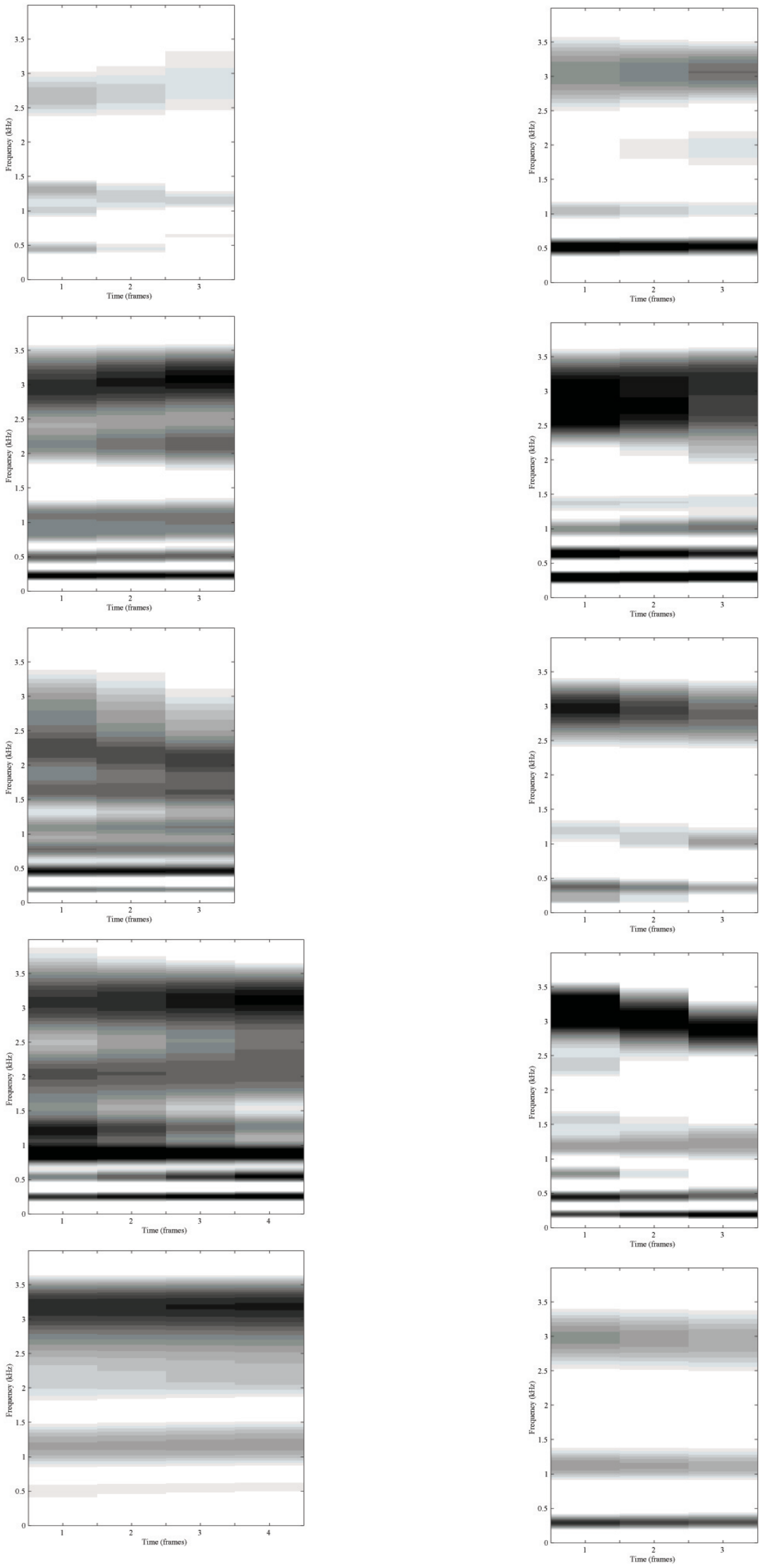

Frequency (kHz)
Time (frames)

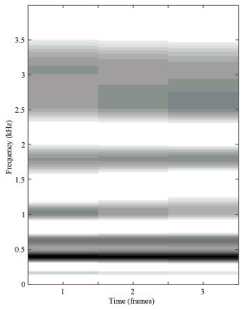
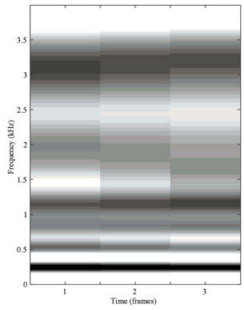
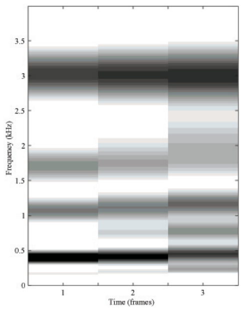
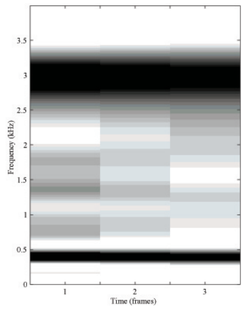
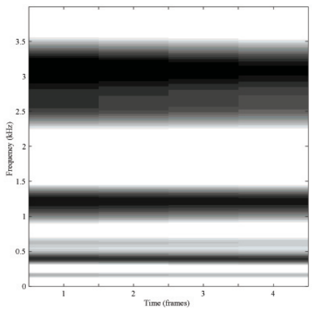
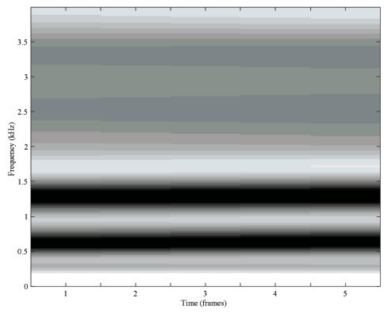
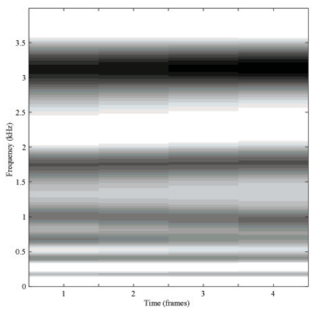
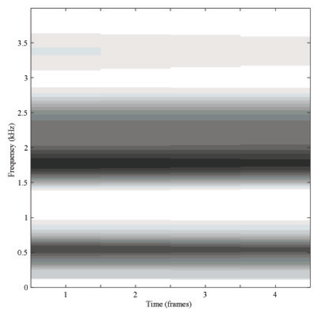
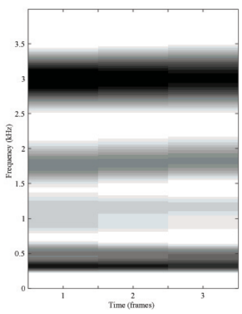
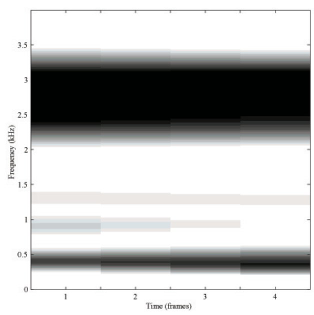

Frequency (kHz)
Time (frames)

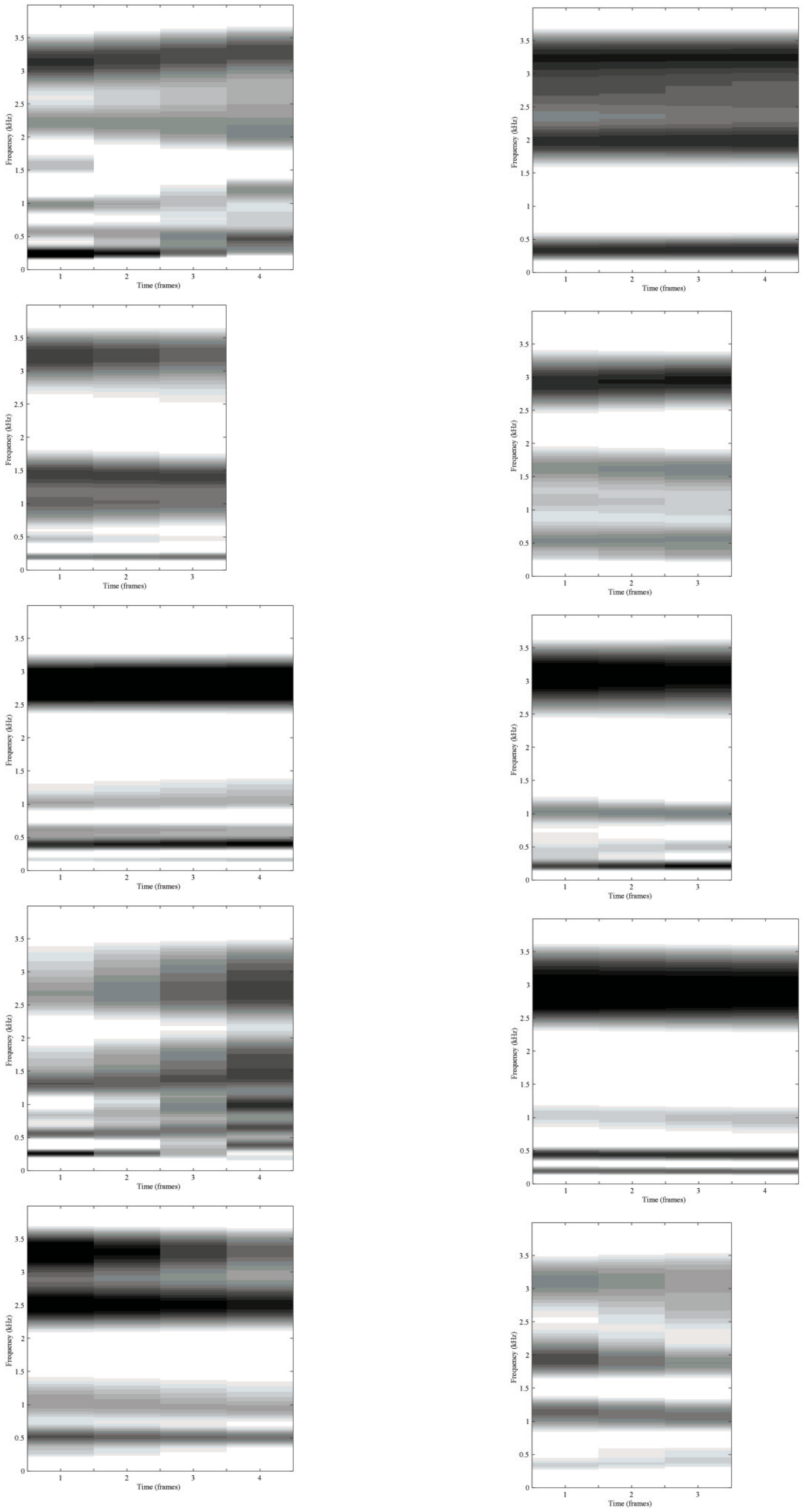

Frequency (kHz)
Time (frames)

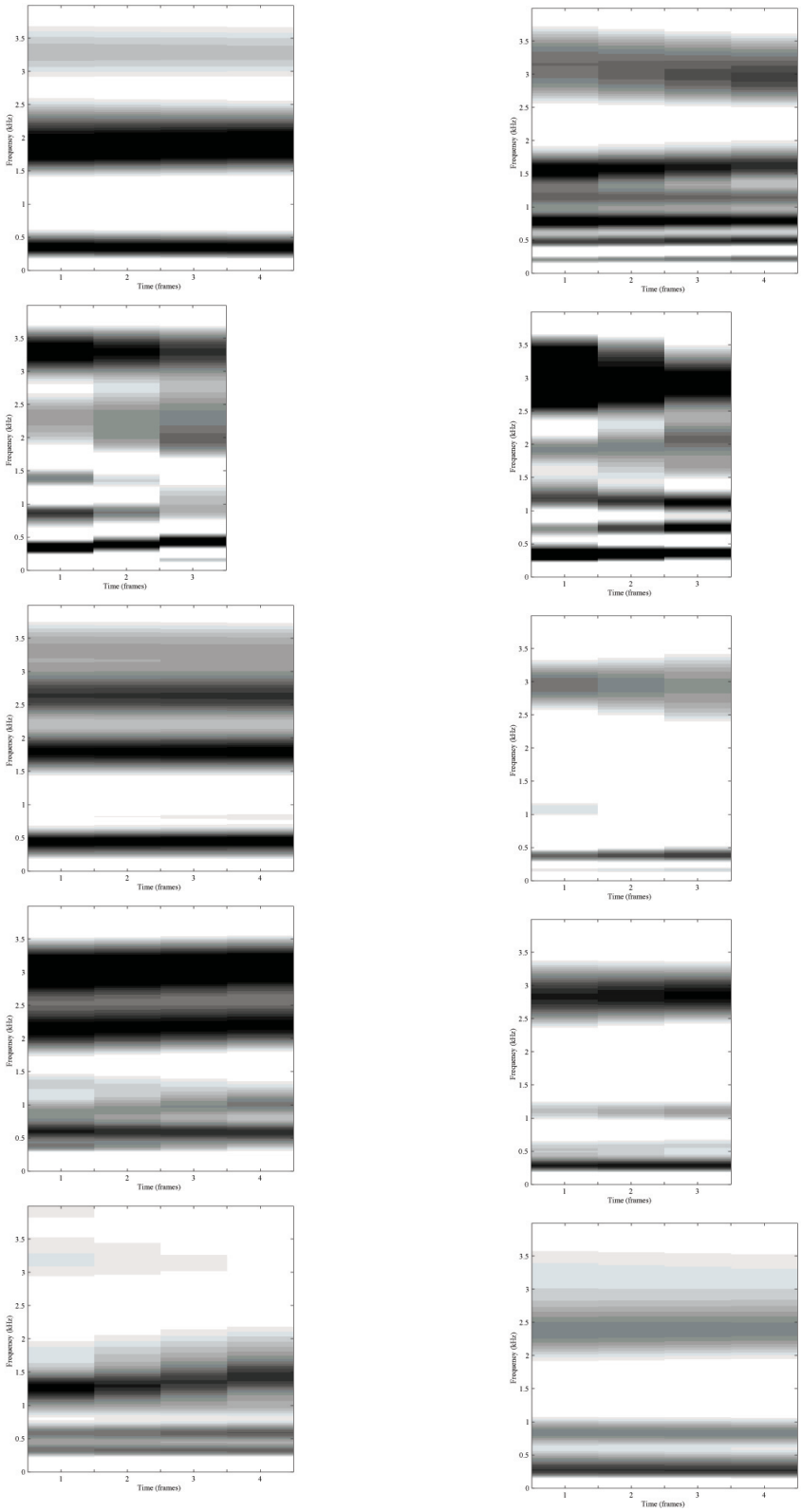

Frequency (kHz)
Time (frames)

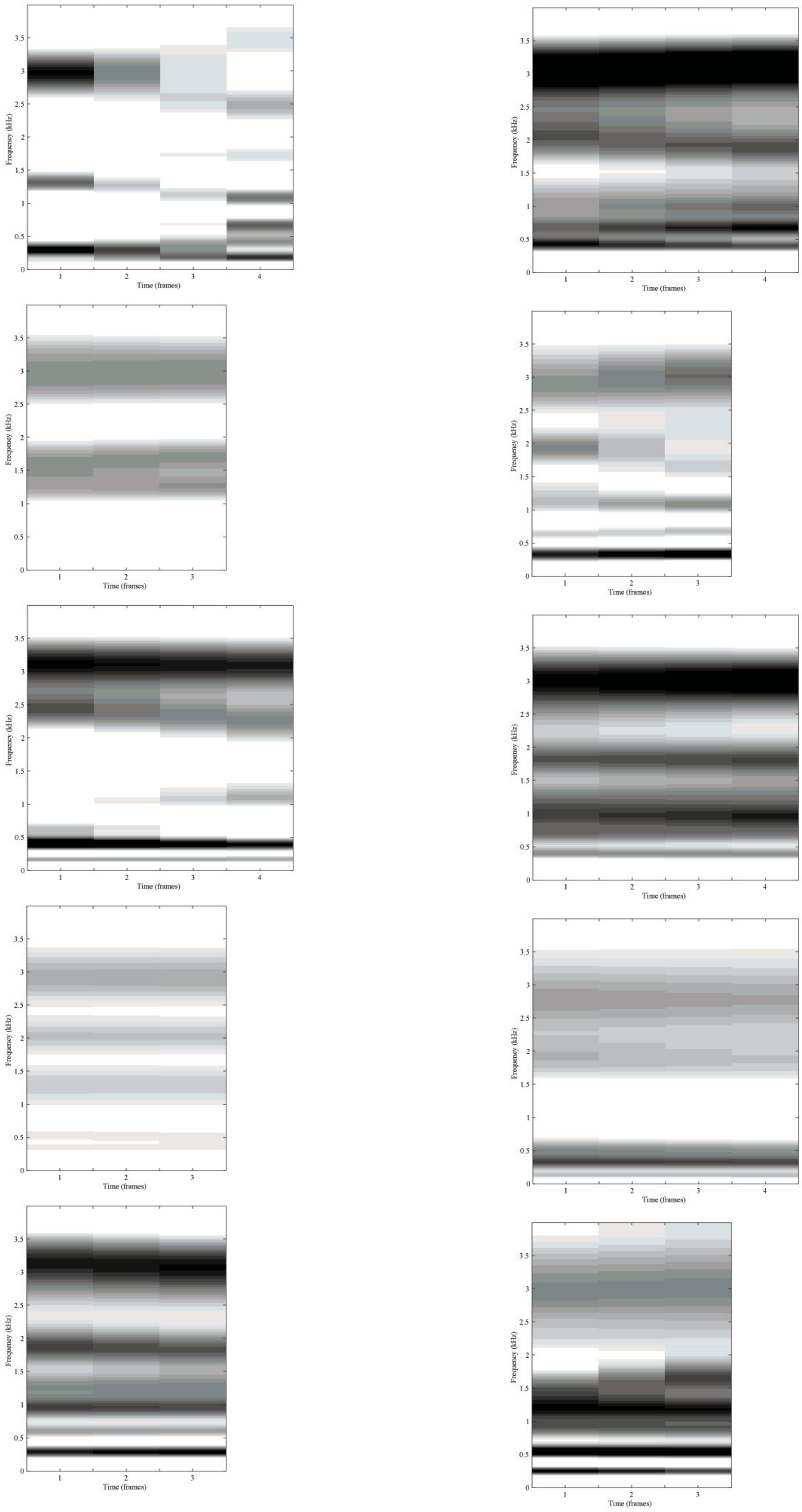

Frequency (kHz)
Time (frames)

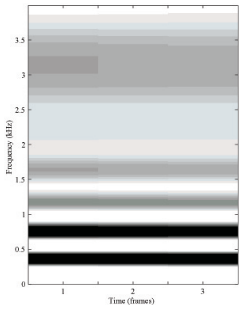

Frequency (kHz)
Time (frames)

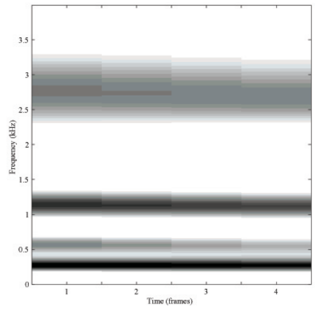

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

Frequency (kHz)
Time (frames)

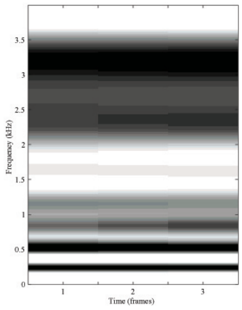

Frequency (kHz)
Time (frames)

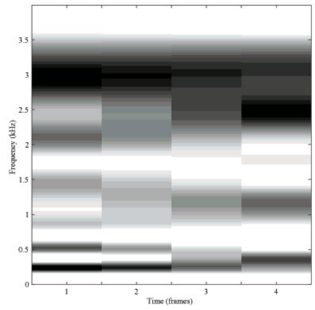

Frequency (kHz)
Time (frames)

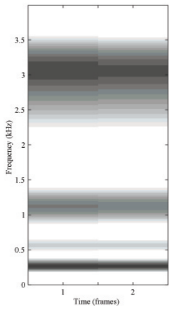

Frequency (kHz)
Time (frames)

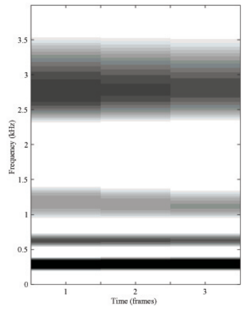

Frequency (kHz)
Time (frames)

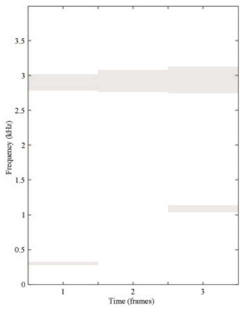

Frequency (kHz)
Time (frames)

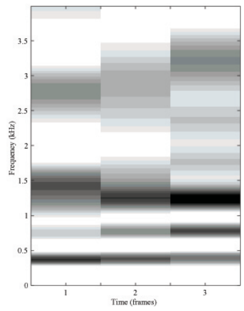

Frequency (kHz)
Time (frames)

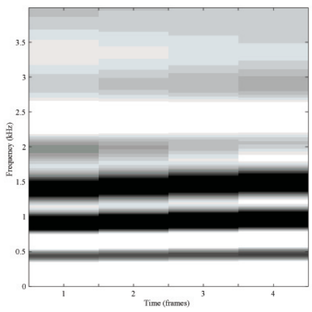

Frequency (kHz)
Time (frames)

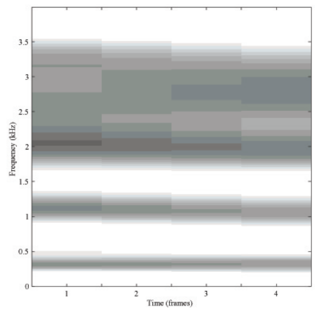

Frequency (kHz)
Time (frames)

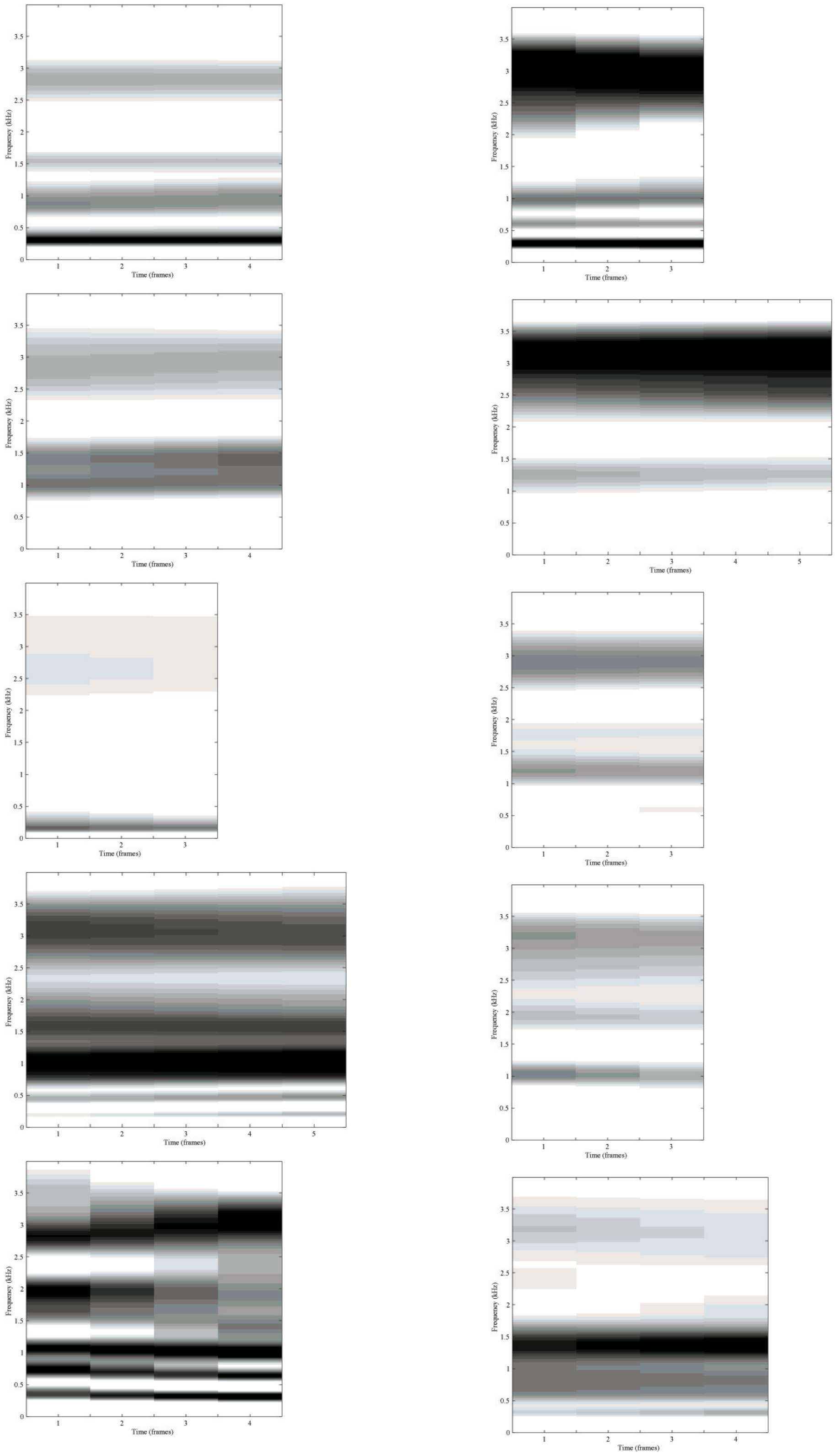

Frequency (kHz)
Time (frames)

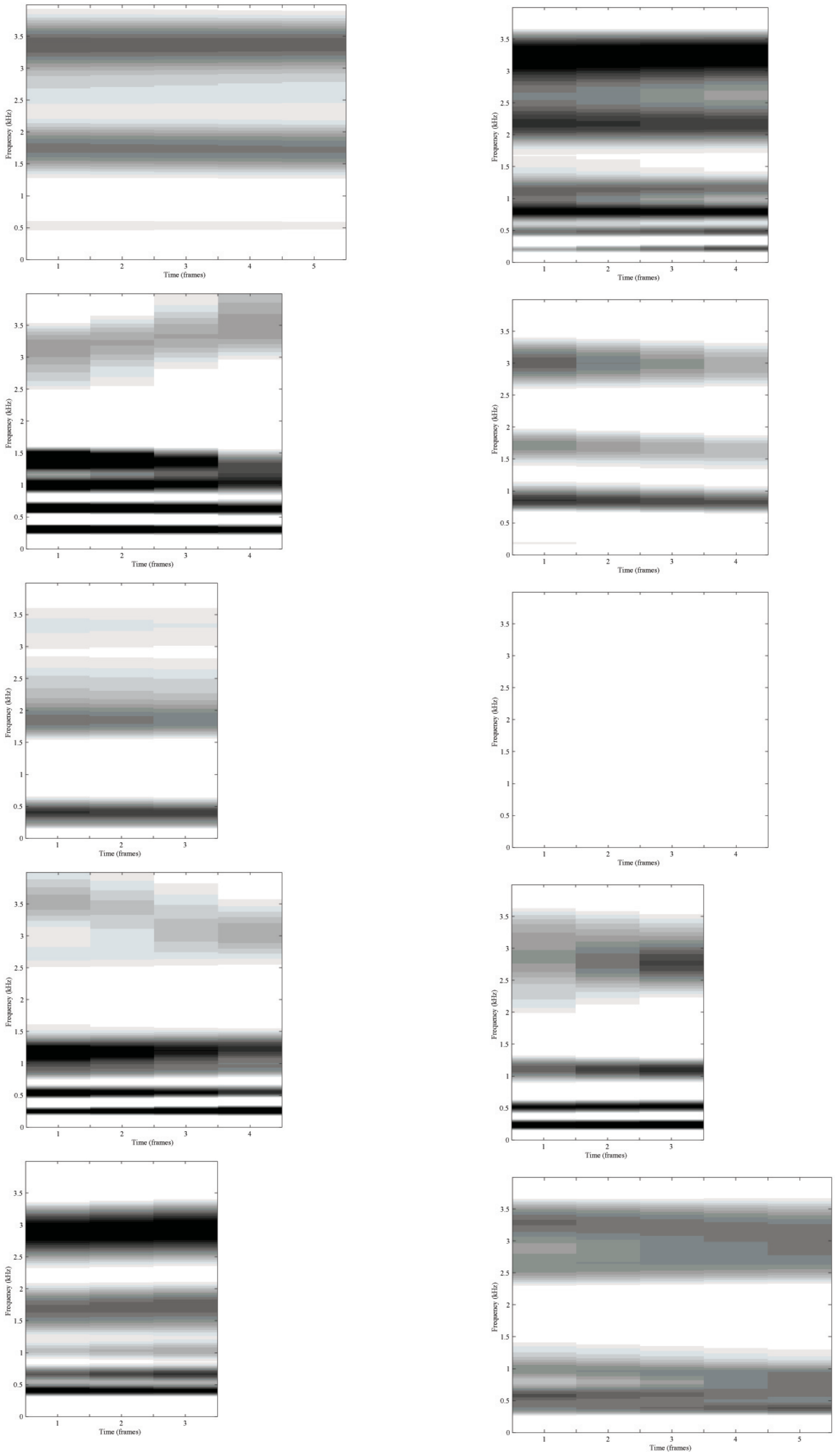

Frequency (kHz)
Time (frames)

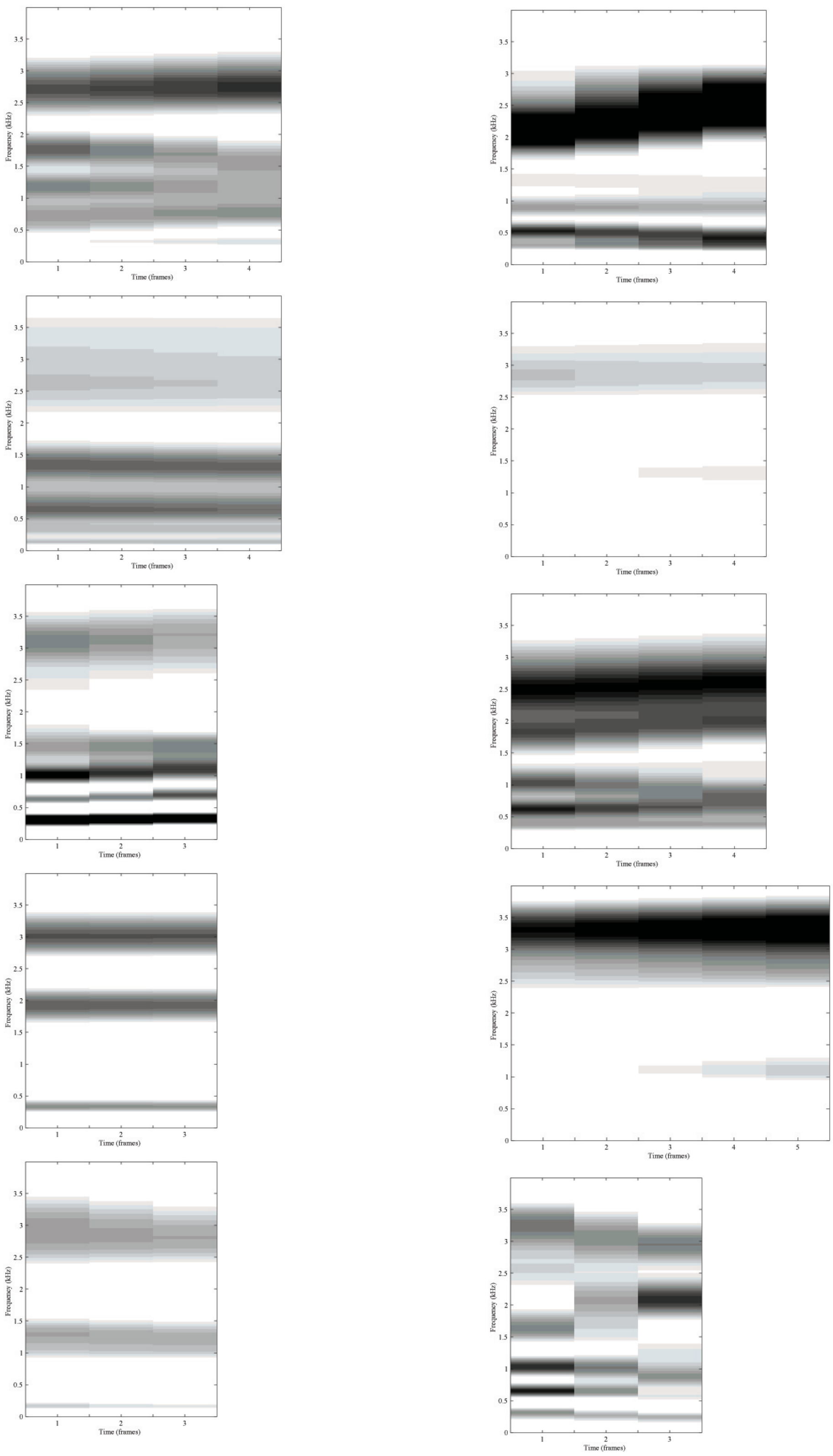

Frequency (kHz)
Time (frames)

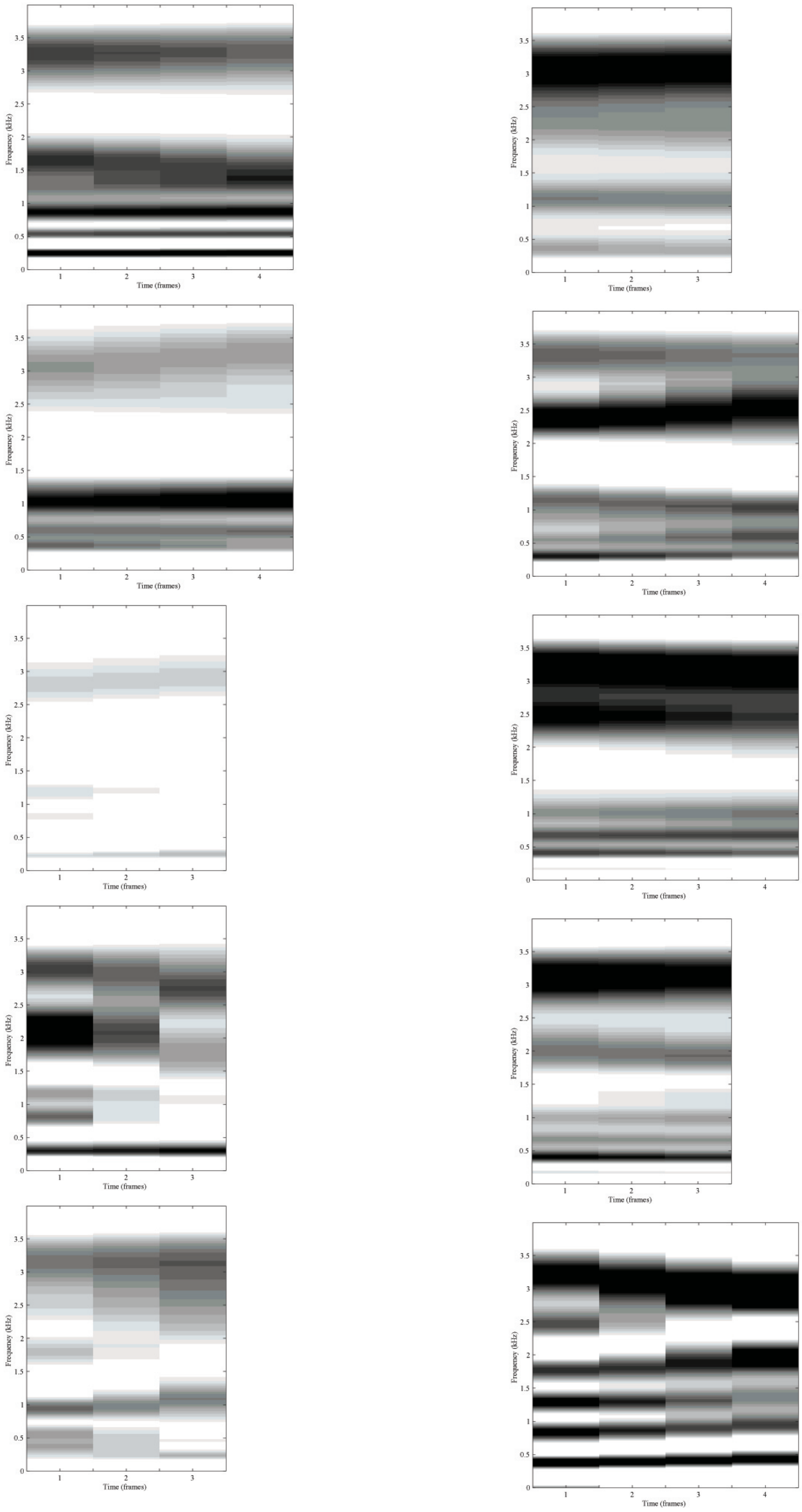

Frequency (kHz)
Time (frames)

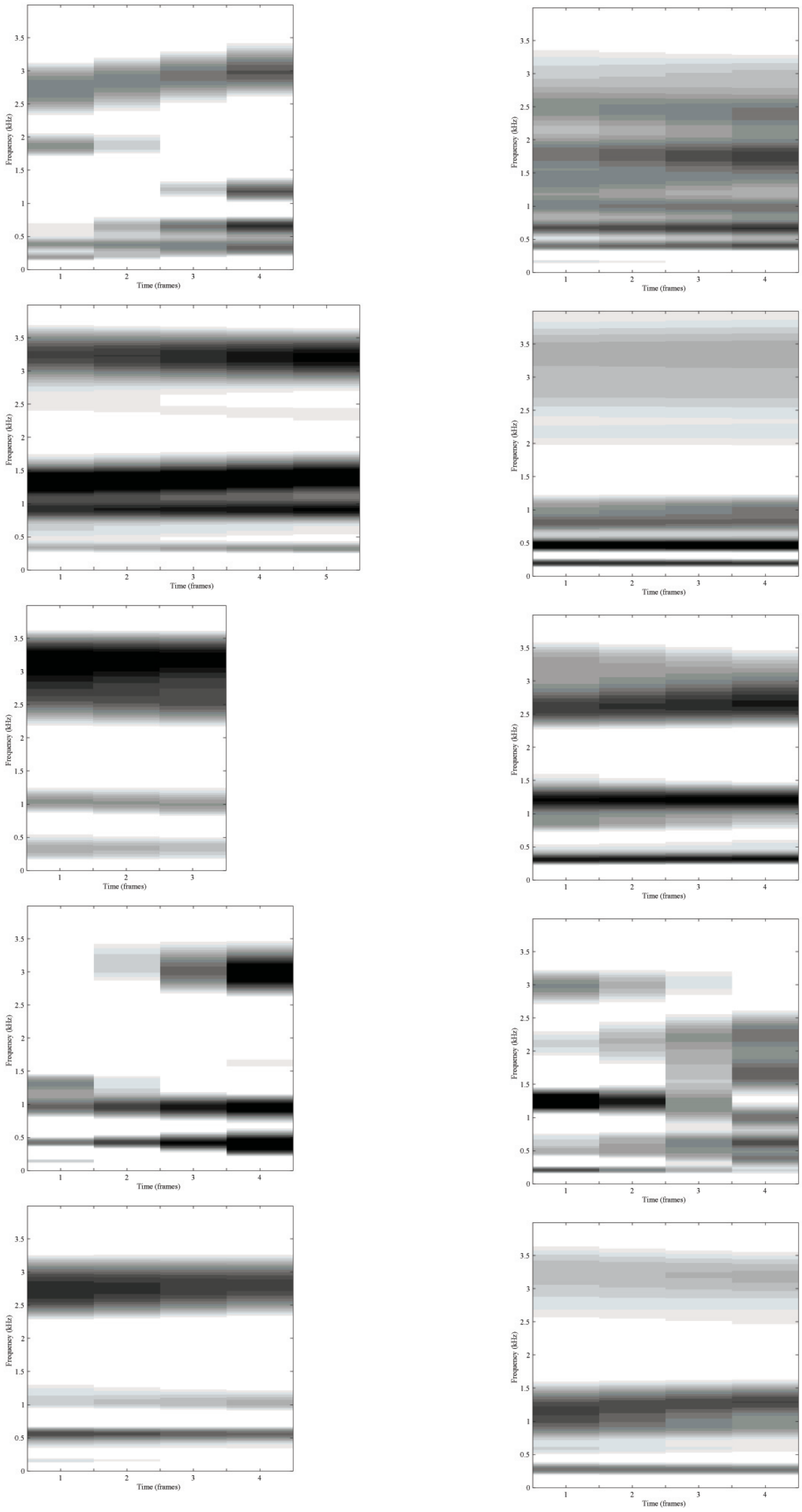

Frequency (kHz)
Time (frames)

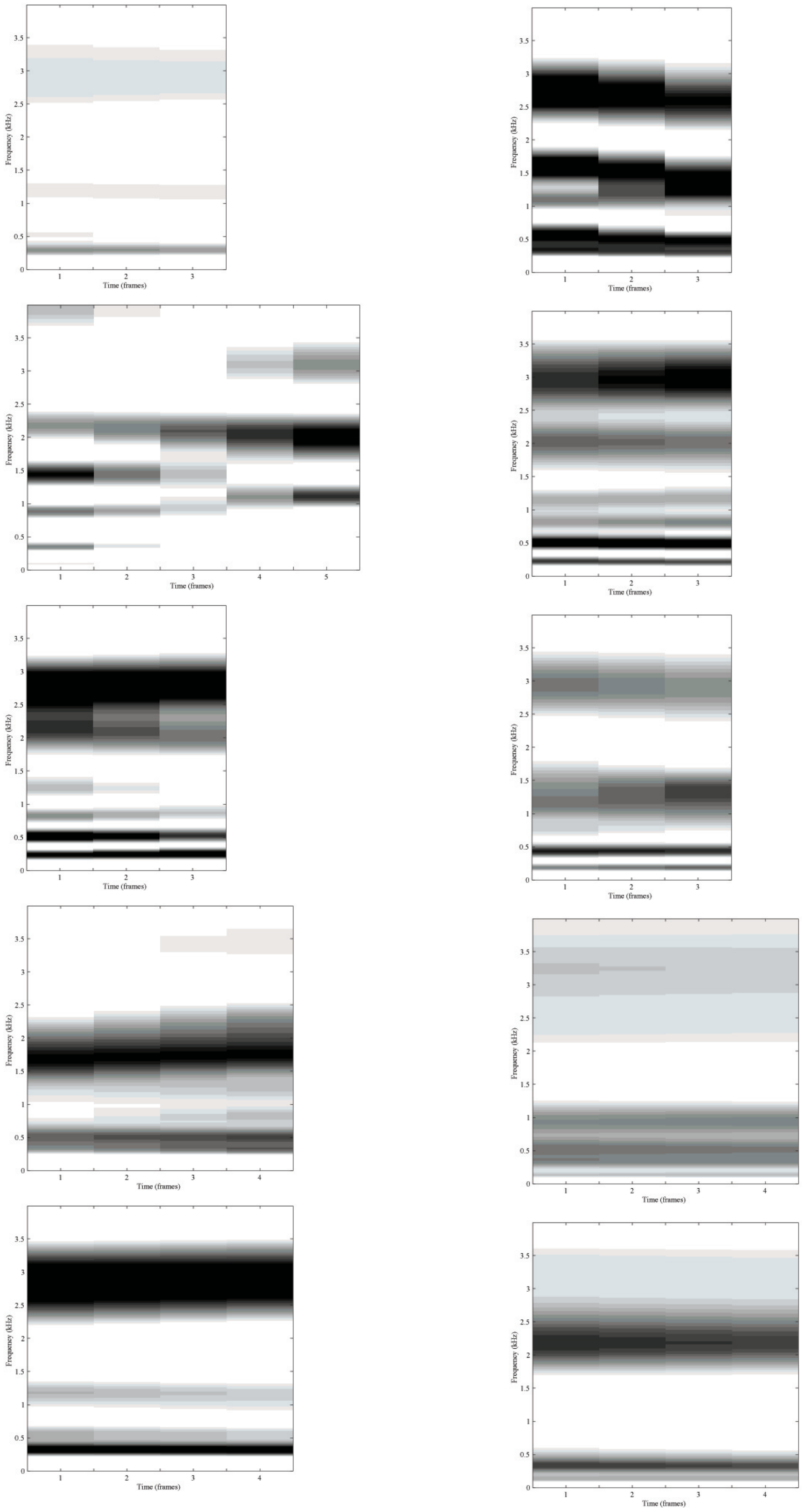

Frequency (kHz)
Time (frames)

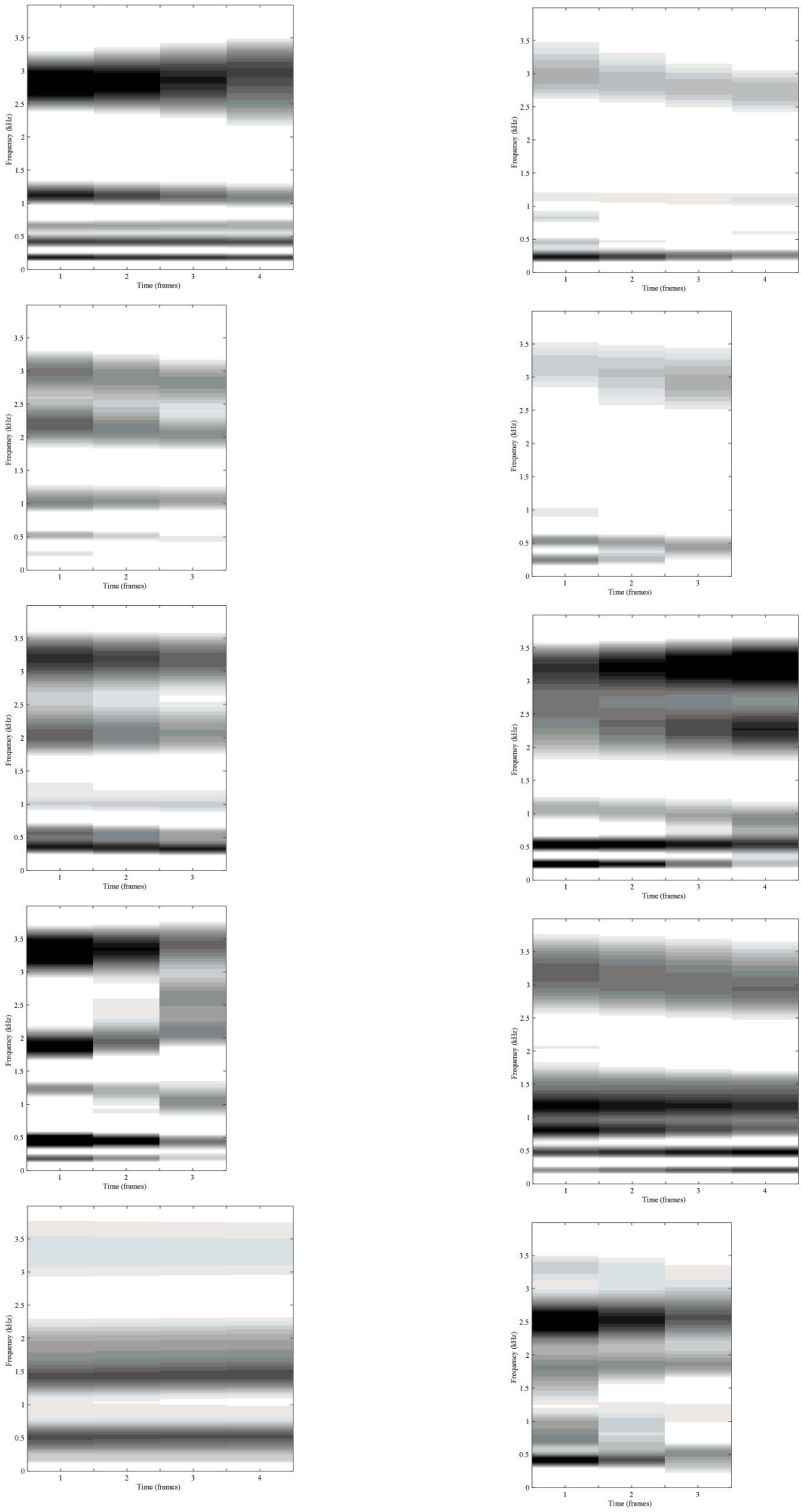

Frequency (kHz)
Time (frames)

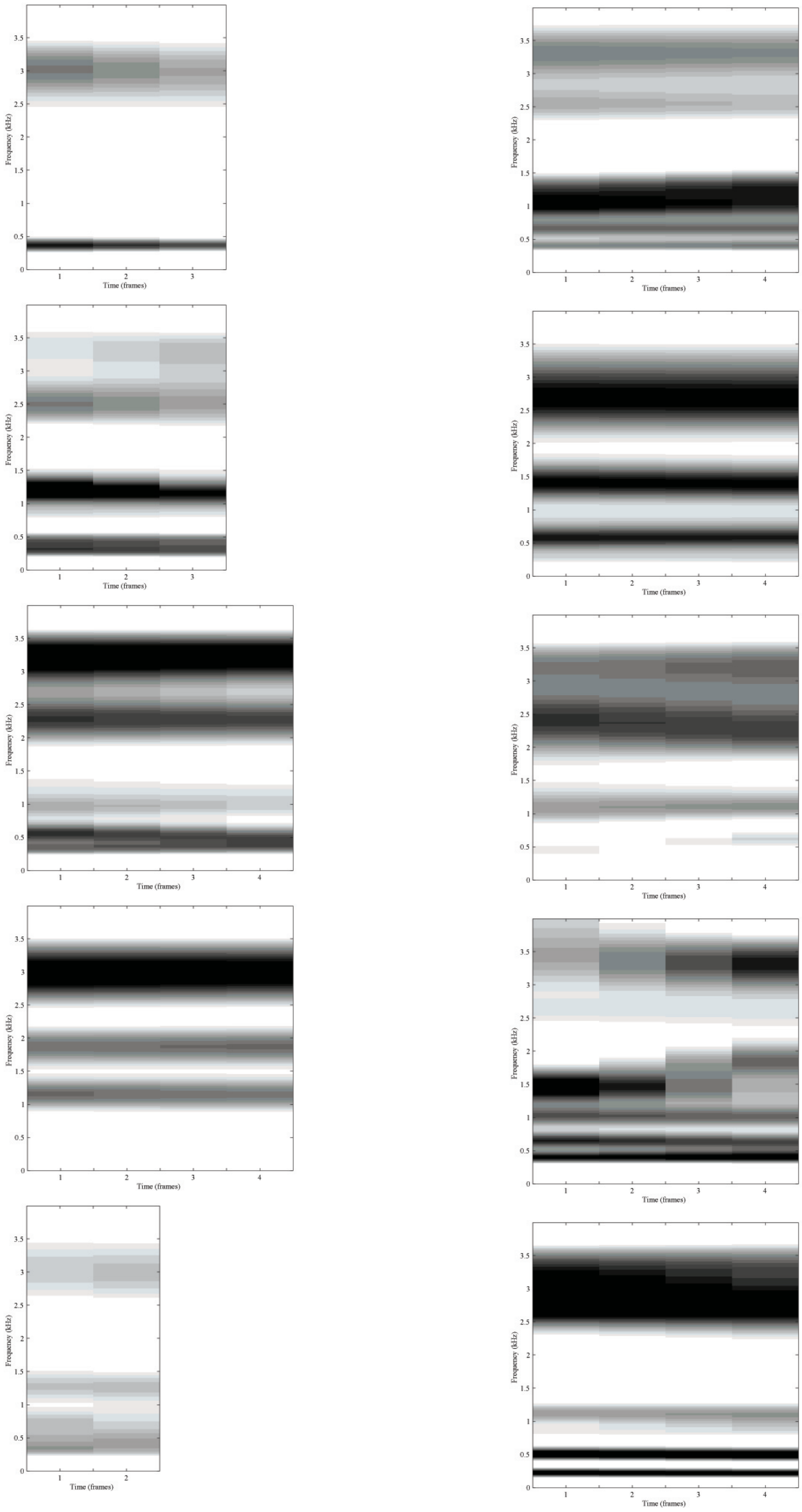

Frequency (kHz)
Time (frames)

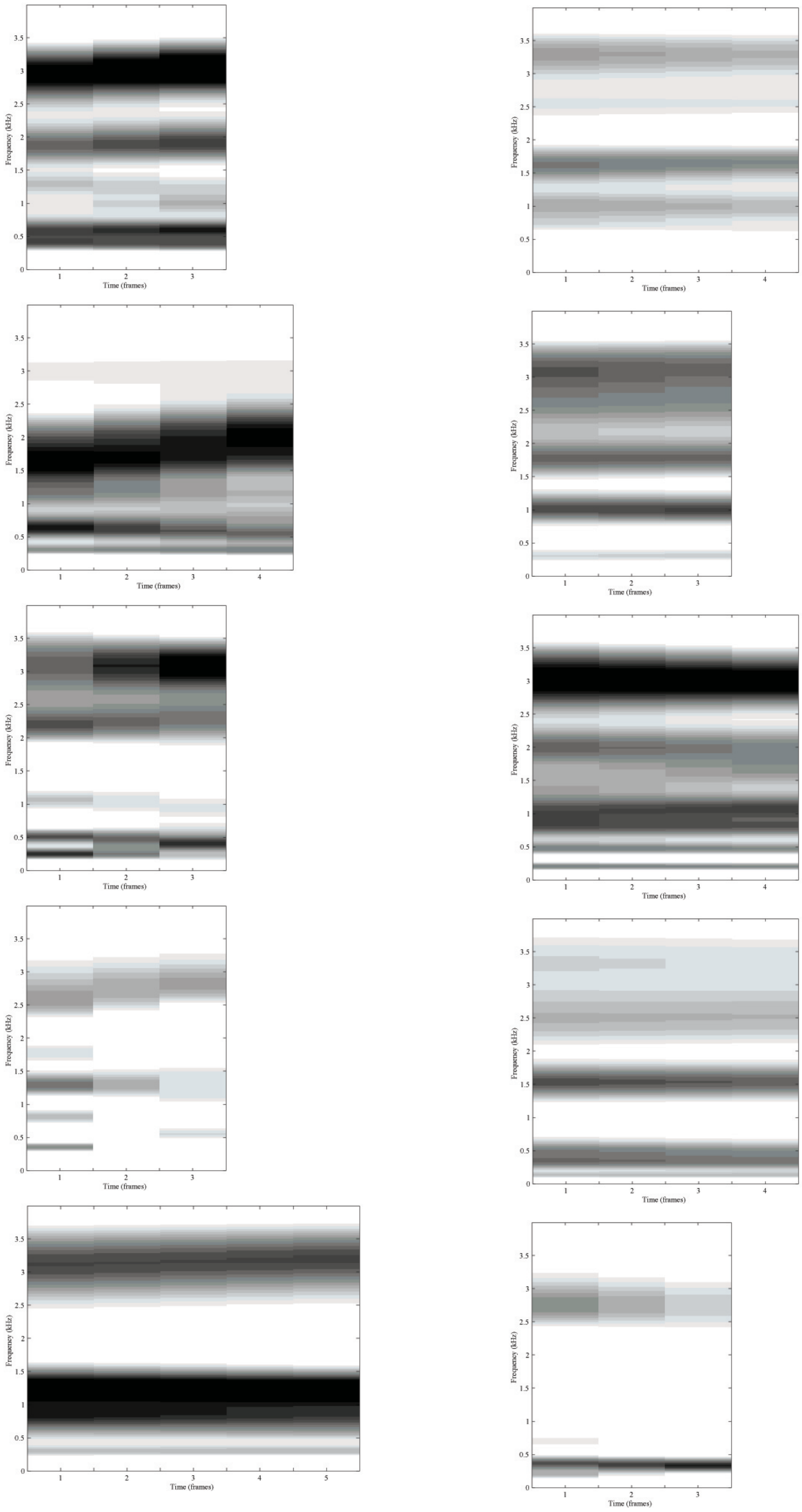

Frequency (kHz)
Time (frames)

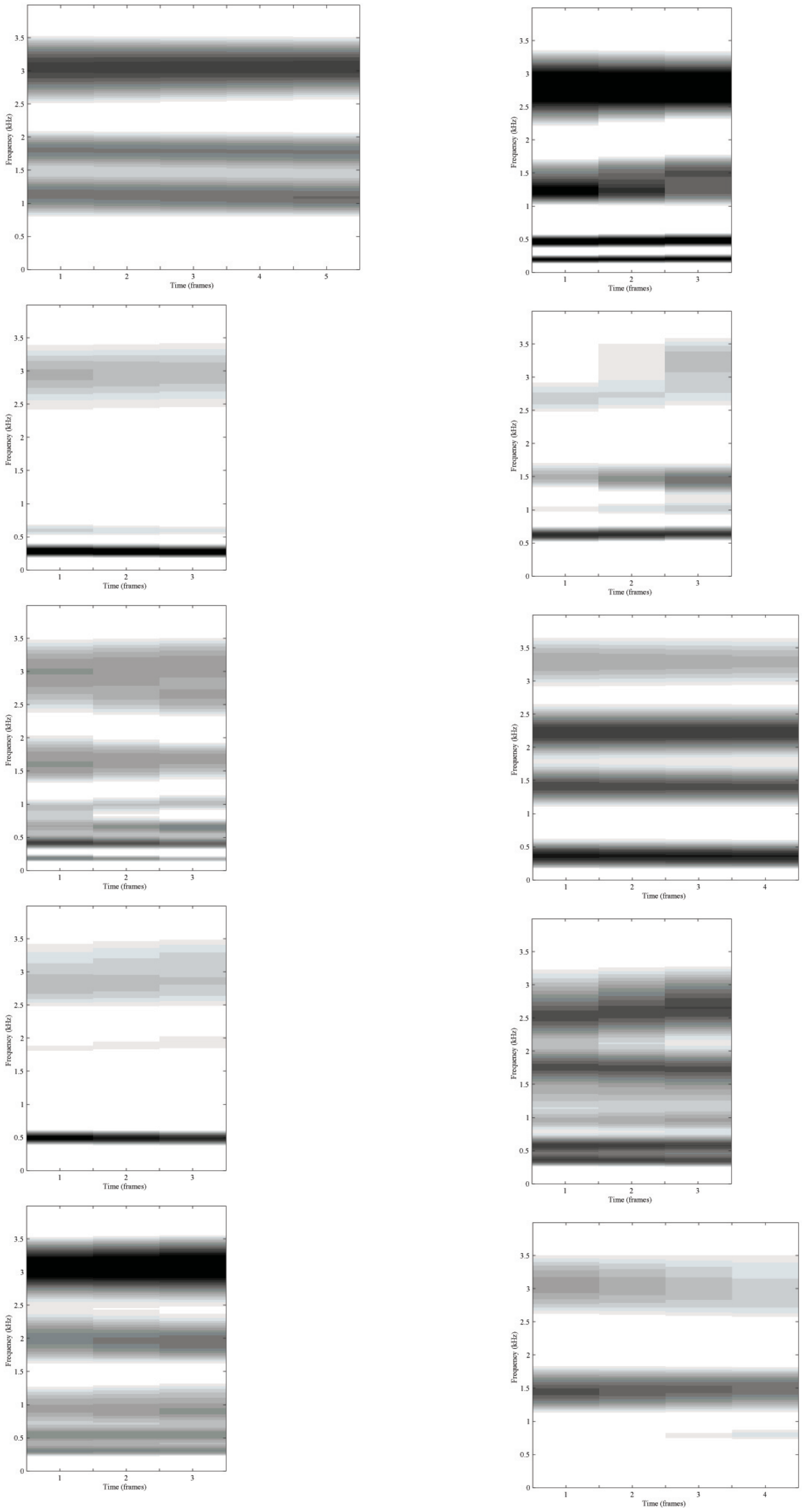

Frequency (kHz)
Time (frames)

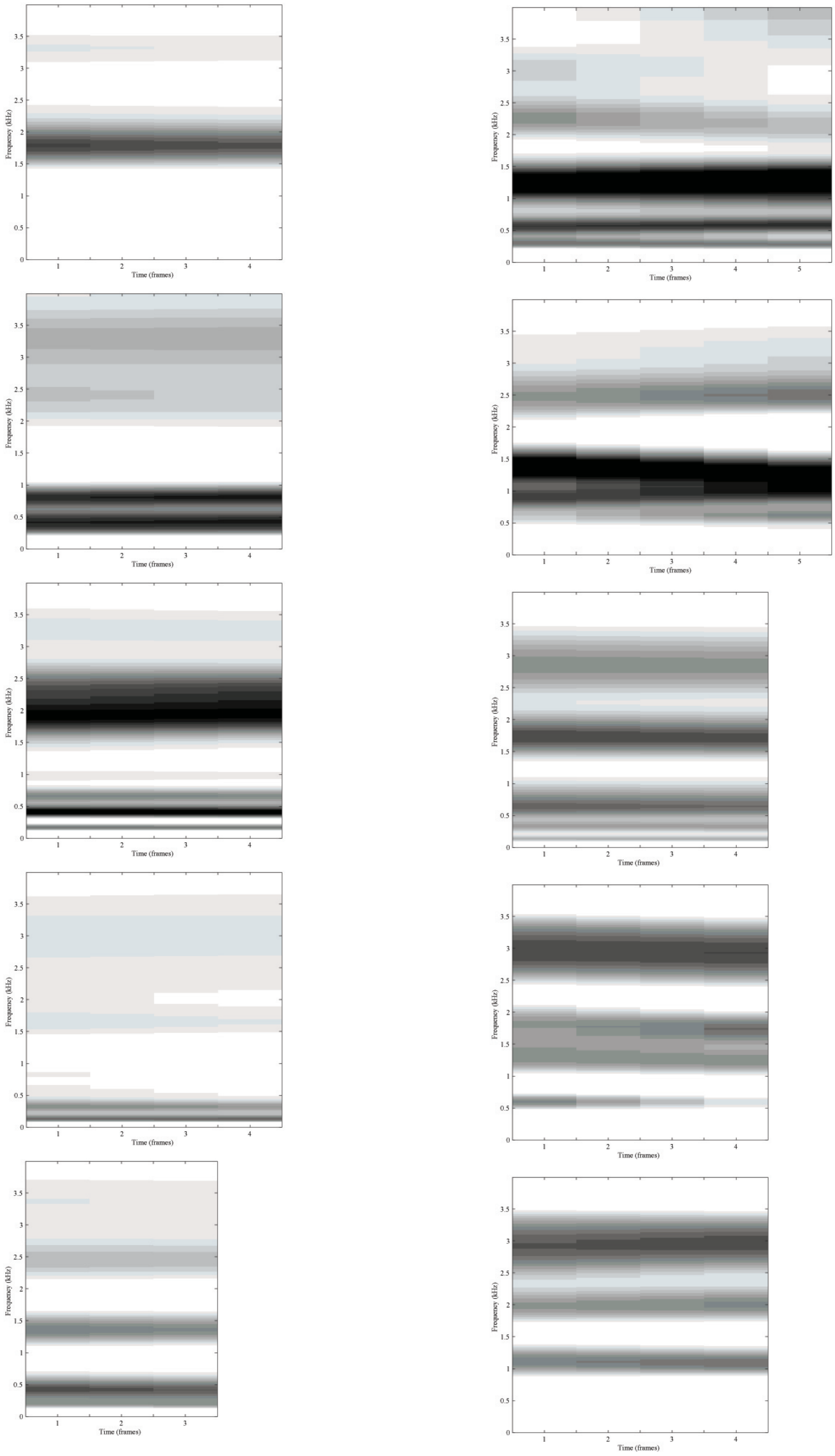

Frequency (kHz)
Time (frames)

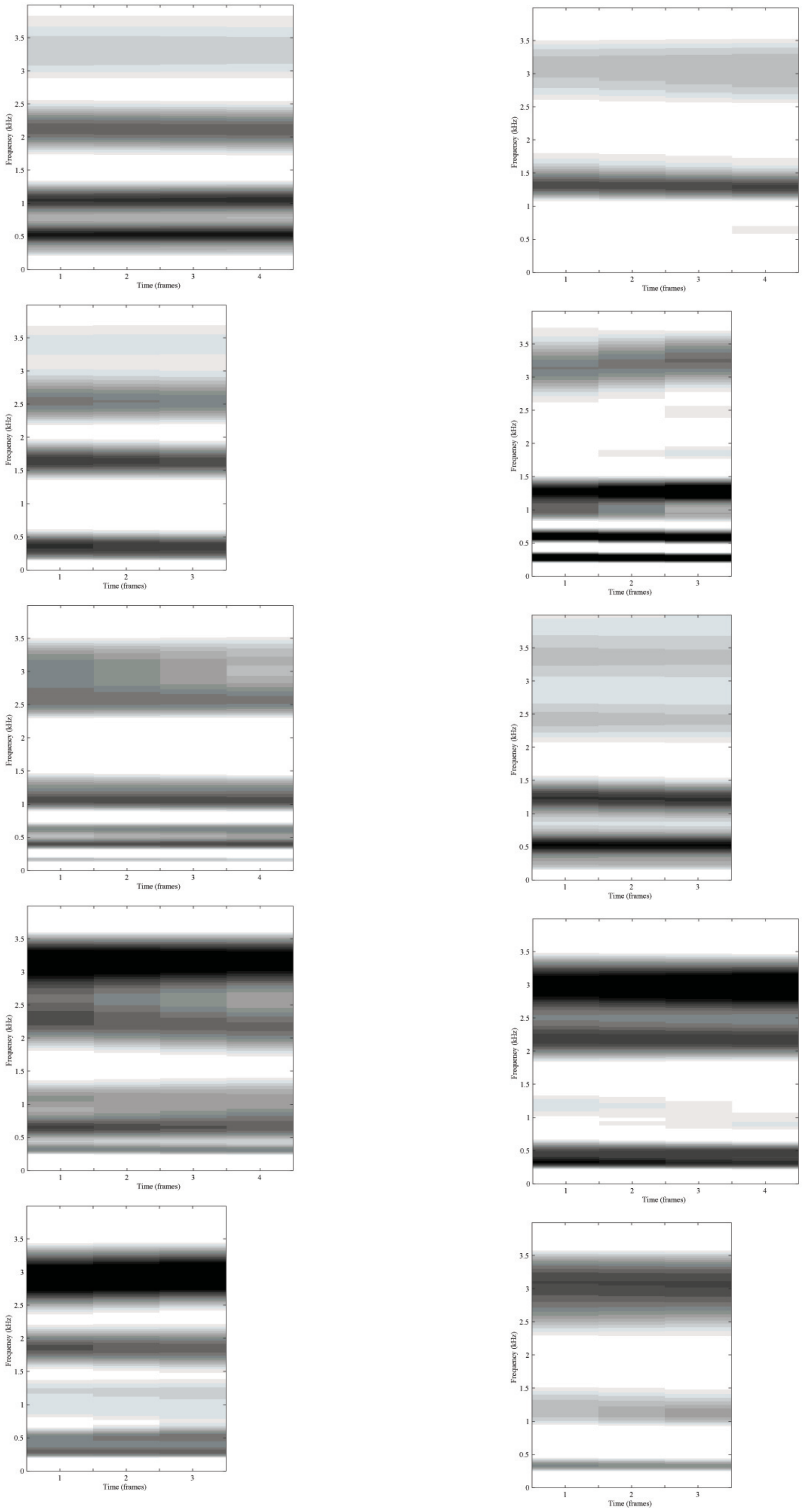

Frequency (kHz)
Time (frames)

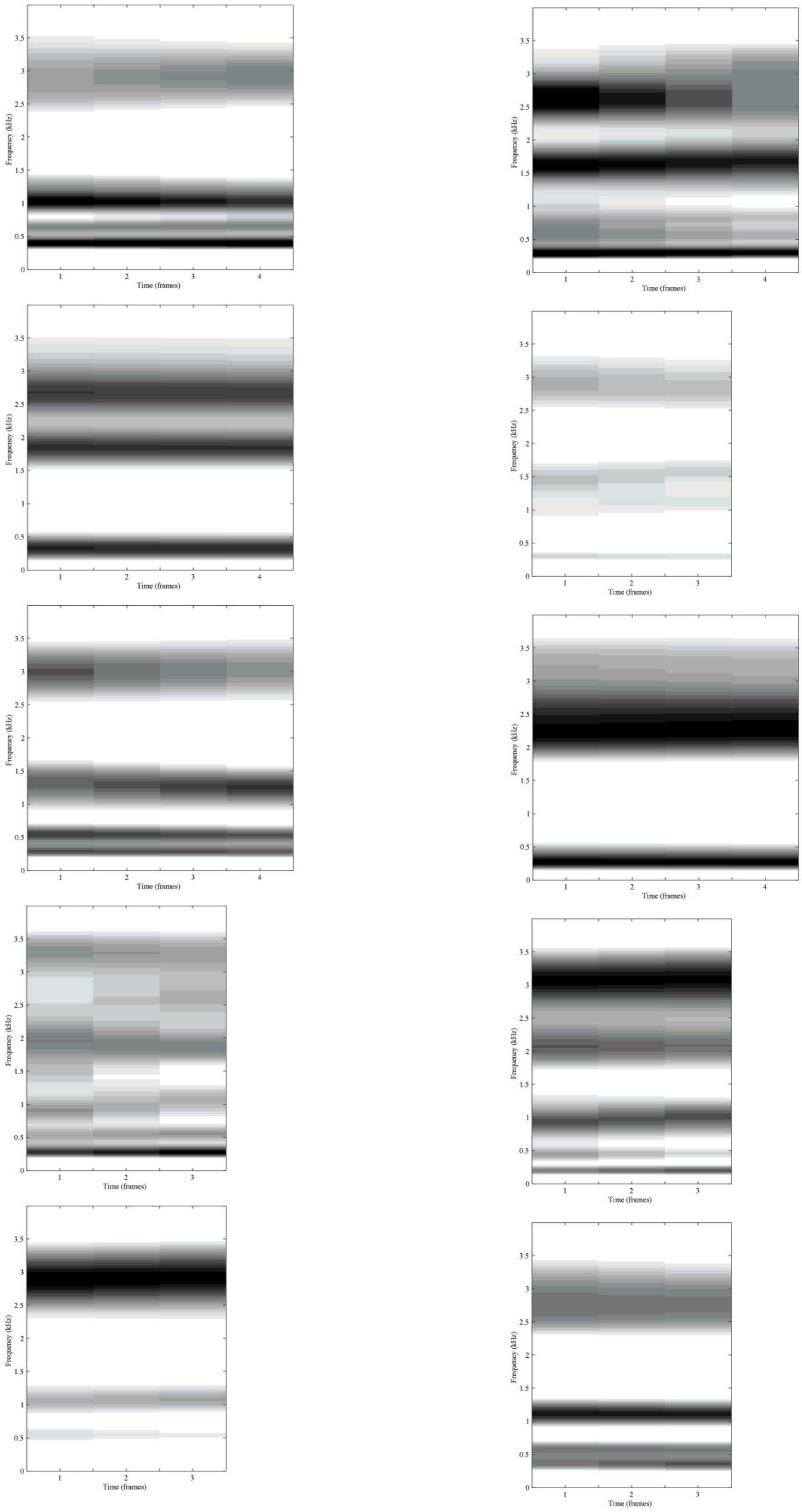

Frequency (kHz)
Time (frames)


Frequency (kHz)
Time (frames)

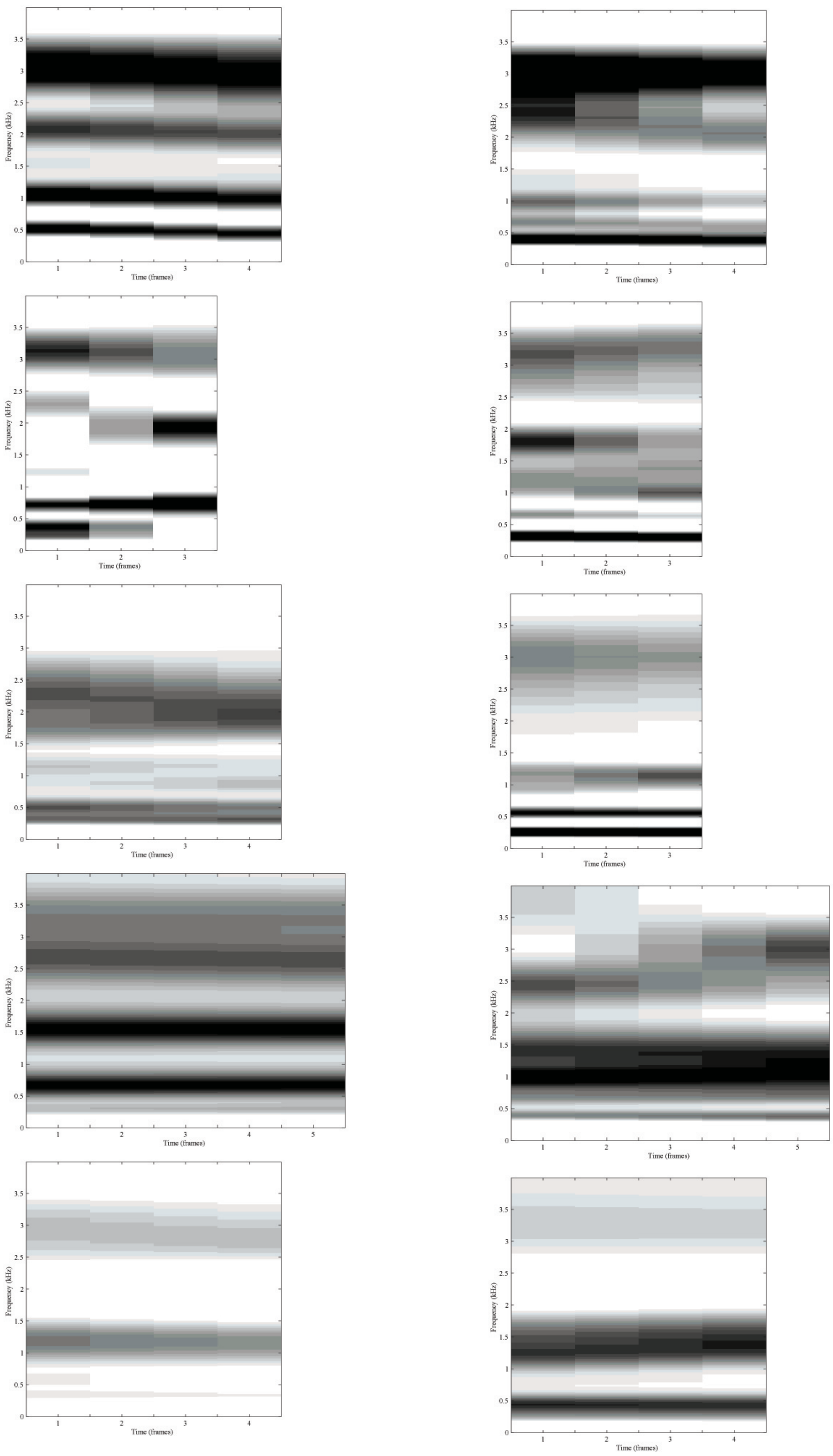

Frequency (kHz)
Time (frames)

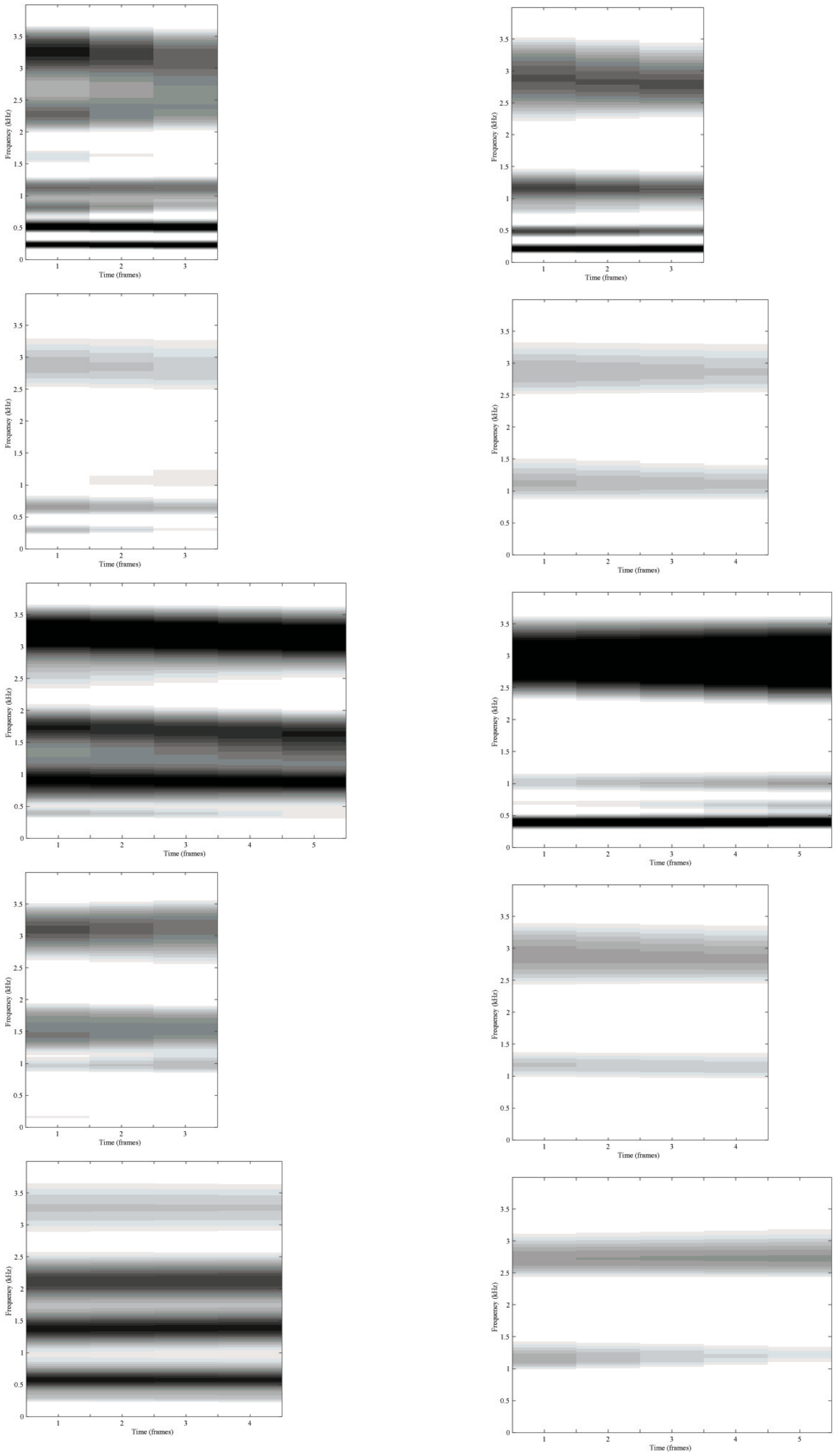

Frequency (kHz)
Time (frames)

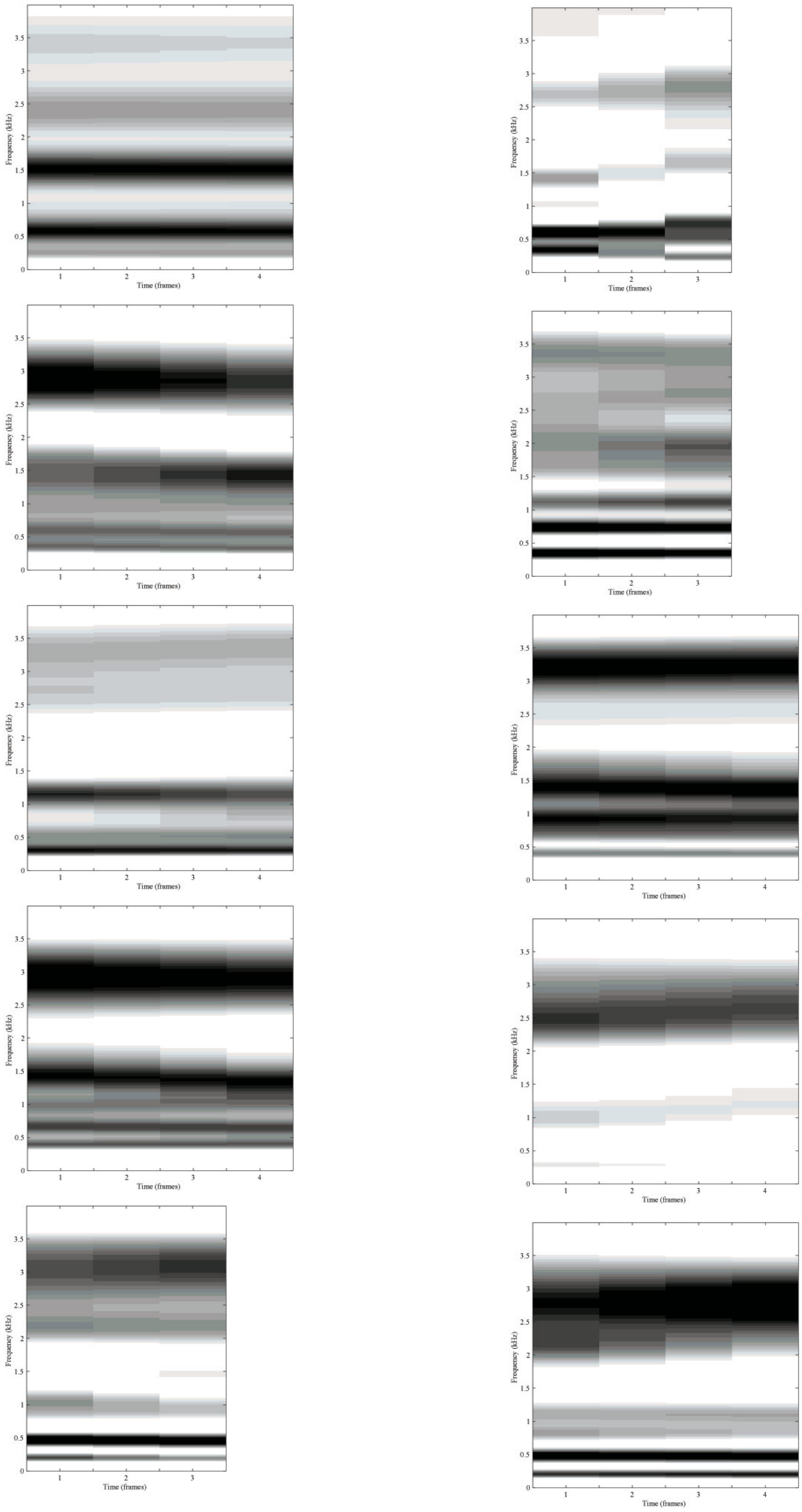

Frequency (kHz)
Time (frames)

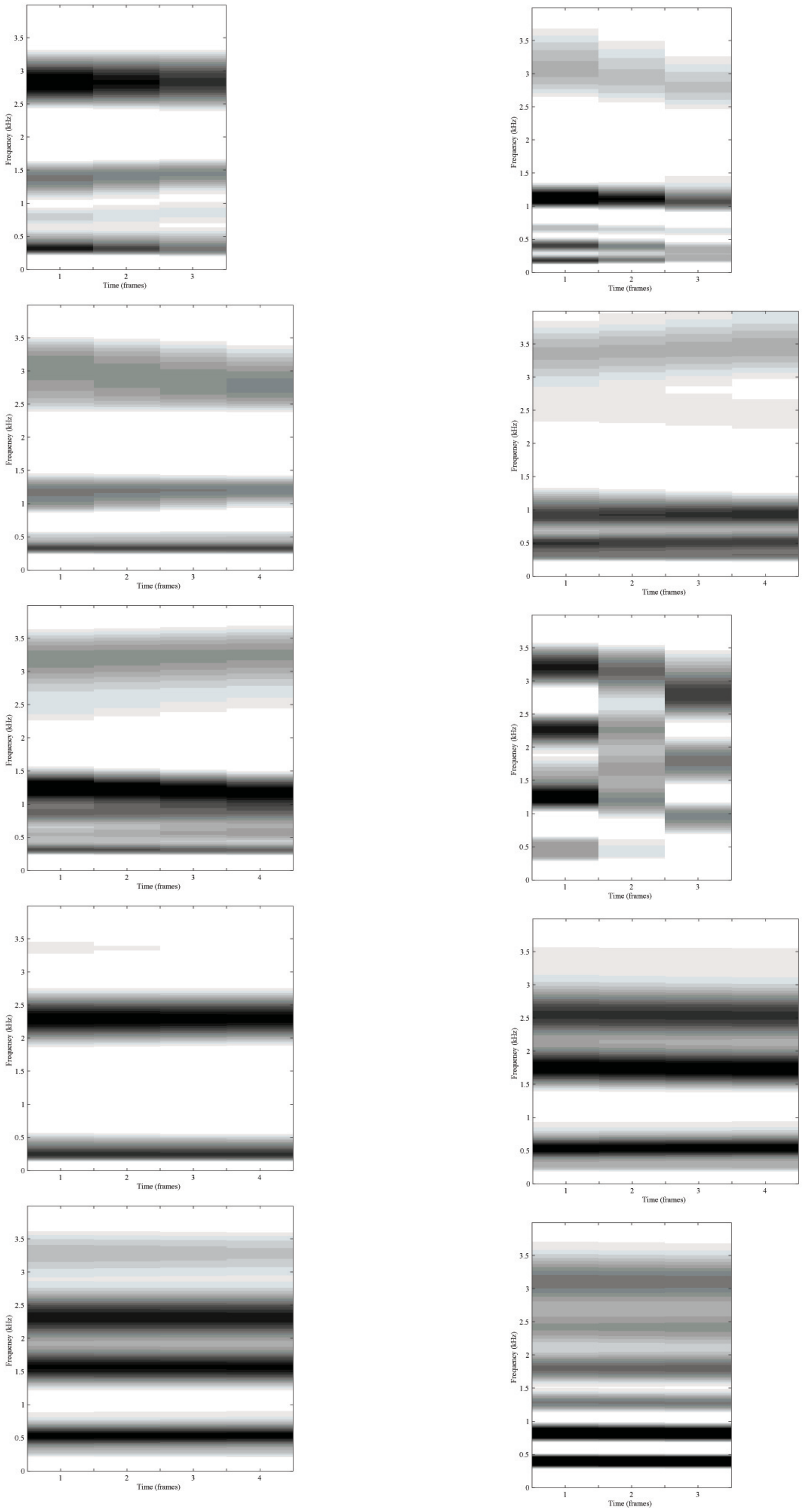


Frequency (kHz)
Time (frames)


Frequency (kHz)
Time (frames)


Frequency (kHz)
Time (frames)


Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
1
2
3
4


Frequency (kHz)
Time (frames)

APPENDIX D

Effects of different number of MFCCs

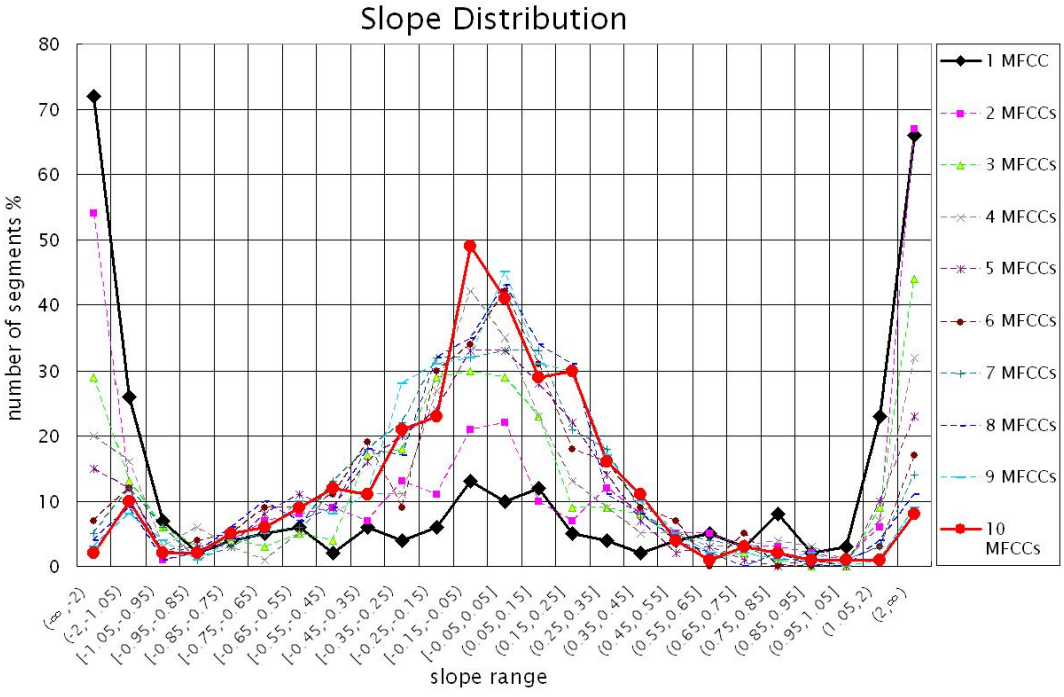


Figure D.1: Distributions of slopes of MFCC_0 in the UBM as the number of MFCCs increases.

APPENDIX E

Publications

"Speaker Recognition Using a Trajectory-Based Segmental HMM", Ying Liu and Martin J. Russell, Proceedings of Odyssey 04, the speaker and language recognition workshop, 2004.

"The Role of Dynamic Features in Text-Dependent and -Independent Speaker Verification", Ying Liu, Martin J. Russell and Michael J. Carey, Proc. ICASSP 2006.

"The Role of 'Delta' Features in Speaker Verification", Ying Liu, Martin J. Russell and Michael J. Carey, Proc. Interspeech 2008.

Bibliography

Adami, A., R. Mihaescu, D. Reynolds, and J. Godfrey (2003). Modeling prosodic dynamics for speaker recognition. In *icassp03*, Volume 4, pp. 788–791.

Ahn, S., S. Kang, and H. Ko (2000). Effective speaker adaptations for speaker verification. In *icassp00*, Volume 2, pp. 1081–1084.

Ainsworth, W. A. (1996). Perceptual tolerance of the shape of formant transitions. *wisp01 18*(9), 67.

Allen, F., E. Ambikairajah, and J. Epps (2005). Language identification using warping and the shifted delta cepstrum. In *Multimedia Signal Processing, 2005 IEEE 7th Workshop on*, pp. 1–4.

Andrews, W. D., M. A. Kohler, J. P. Campbell, J. J. Godfrey, and J. Hernandez-Cordero (2002). Gender-dependent phonetic refraction for speaker recognition. In *icassp02*.

Appiah, M., M. Sasikath, R. Makrickaite, and M. Gusaite (2005). *Robust*

Voice Activity Detection and Noise Reduction Mechanism. Institute of Electronics Systems, Aalborg University.

Atal, B. S. (1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *jasa 51-A*, 34–42.

Auckenthaler, R. (2001). *Text-independent speaker verification with limited resources*. Ph. D. thesis, University of Wales Swansea.

Auckenthaler, R., M. J. Carey, and H. Lloyd-Thomas (2000, January/April/July). Score normalisation for text-independent speaker verification systems. *Digital Signal Processing 10*(1-3), 42–54.

Bahl, L. R., F. Jelinek, and R. L. Mercer (1983). A maximum likelihood approach to continuous speech recognition. *PAMI-5*(2), 179–190.

Bielefeld, B. (1994). Language identification using shifted delta cepstrum. In *Proc. Fourteenth Annual Speech Research Symposium*.

Bridle, J. S. (2004). Towards better understanding of the model implied by the use of dynamic features in hmms. In *icslp04*.

Brown, P. F. (1987). *The acoustic modeling problem in automatic speech recognition*. Ph. D. thesis, Carnegie Mellon University.

Burget, L., P. Matejka, P. Schwarz, O. Glembek, and J. Cernocky (2007). Analysis of feature extraction and channel compensation in a gmm speaker recognition system. In *ieeeaslp*, Volume 15, pp. 1979–1986.

Calvo, J. R., R. Fernandez, and G. Hernandez (2007). Application of shifted delta cepstral features in speaker verification. In *interspeech07*,

pp. 734–737.

Campbell, W., J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo (2006). Support vector machines for speaker and language recognition. *csl 20*(2-3), 210–229.

Campbell, W. and D. Reynolds (1999). Corpora for the evaluation of speaker recognition systems. In *ieeeaslp*, Volume 2, pp. 829–832.

Campbell, W., D. Sturim, D. Reynolds, and A. Solomonoff (2006). Svm based speaker verification using a gmm supervector kernel and nap variability compensation,. In *icassp06*, pp. 97–100.

Campbell, W. M. (2002). Generalized linear discriminant sequence kernels for speaker recognition. In *icassp02*, pp. 161–164.

Carey, M. J., E. Parris, and J. Bridle (1991). A speaker verification system using alpha-nets. In *icassp91*, pp. 397–400.

Cole, R., M. Noel, and V. Noel (1998). The cslu speaker recognition corpus. In *icslp98*, pp. 3167–3170.

Consortium, I. D. (1992). *King kingdb.doc and collectn.doc files*. Univ. Pennsylvania, Philadelphia, PA: Linguistic Data Consortium.

Davis, S. B. and P. Mermelstein (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *ASSP-28*, 357–366.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*(39), 1–38.

Digalakis, V. (1992). *Segment-based stochastic models of spectral dynamics for continuous speech recognition*. Ph. D. thesis, Boston University, MA.

Doddington, G. (2001). Speaker recognition based on idiolectal differences between speakers. In *euro01*, Volume 4, pp. 2517–2520.

Fant, G. (1960). *Acoustic Theory of Speech Production*. Mouton: The Hague.

Farrell, K. R., R. J. Mammone, and K. T. Assaleh (1994). Speaker recognition using neural networks and conventional classifiers. *ieeesap 2*, 194–205.

Ferguson, J. D. (1980). Hidden markov analysis. *in Hidden Markov Models for Speech, Institute for Defense Analysis, Princeton, NJ*.

Fine, S., J. Navratil, and R. A. Gopinath (2001). A hybird gmm/svm approach to speaker recognition. In *icassp01*.

Forney, G. (1973). The viterbi algorithm. In *ieeep*, Volume 61, pp. 268–278.

Furui, s. (1981). Cepstral analysis technique for automatic speaker verification. *icassp81 29*(2), 254–272.

Gales, M. J. F. and S. J. Young (1992). An improved approach to the hidden markov model decomposition of speech and noise. In *icassp92*, Volume 1, pp. 233–236.

Gales, M. J. F. and S. J. Young (1993). Segmental hidden markov models. In *euro93*, pp. 1579–1582.

Ganapathiraju, A. and J. Picone (2000). Hybrid svm/hmm architectures for speech recognition. In *Speech Transcription Workshop*.

Garofolo et al., J. S. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Univ. Pennsylvania, Philadelphia, PA: Linguistic Data Consortium.

Gauvain, J.-L. and C. Lee (1994). Maximum a-posteriori estimation for multivariate gaussian mixture observations of markov chains. *2*, 291–298.

Ghitza, O. and M. Sondhi, M (1993). Hidden markov models with templates as non-stationary states: an application to speech recognition. *csl 2*, 101–119.

Gish, H. and K. Ng (1993). A segmental speech model with applications to word spotting. In *icassp93*, pp. 447–450.

Greenberg, S. and B. Kingsbury (1997). The modulation spectrogram: in pursuit of an invariant representation of speech. In *icassp97*, Volume 3, pp. 1647–1650.

Hansen, E. G., R. E. Slyh, and T. R. Anderson (2004). Speaker recognition using phoneme-specific gmms. In *odyssey04*, pp. 179–184.

Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete fourier transform. In *ieeep*, Volume 66, pp. 51–83.

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *jasa* (87), 1738–1752.

Hermansky, H. and N. Morgan (1994). Rasta processing of speech. In *ieeesap*, Volume 2.

Higgins, A. (1990). Yoho speaker verification. In *Presented at the Speech*

Research Symposium, Baltimore, MD.

Holmes, W. J. and M. J. Russell (1996). Modelling speech variability with segmental hmms. In *icassp96*, pp. 447–450.

Holmes, W. J. and M. J. Russell (1997). Linear dynamic segmental hmms: Variability representation and training procedure. In *icassp97*, pp. 1399–1402.

Holmes, W. J. and M. J. Russell (1999). Probablistic-trajectory segmental HMMs. *csl 13*(1), 3–37.

Jaakkola, T. and D. Haussler (1998). Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing*, Volume 11, pp. 487–493.

Jackson, P. J. B. and M. J. Russell (2002). Models of speech dynamics in a segmental-HMM recogniser using intermediate linear representations. In *icslp02*, pp. 1253–1256.

Jelinek, F. (1976, April). Continuous speech recognition by statistical methods. In *ieeep*, Volume 64, pp. 532–556.

Jin, J.-H., M. J. Russell, M. J. Carey, J. Chapman, H. Lloyd-Thomas, and G. D. Tattersall (2003). A spoken language interface to an electronic programme guide. In *euro03*.

Juang, B. (1984). On the hidden markov model and dynamic time warping for speech recognition - a unified view. *AT&T B. L. T. J. / 63*(7), 1213–1243.

Juang, B. H. and L. R. Rabiner (1991). Hidden markov models for speech

recognition. In *Technometrics*, Volume 33, pp. 251–272.

Kenny, P., M. Lennig, and P. Mermelstein (1990). A linear predictive hmm for vector-valued observations with applications to speech recognition. In *IEEE Trans. Acoust. Speech, Signal Processing*, Volume 38, pp. 220–225.

Kharroubi, J., D. Petrovska-Delacretaz, and G. Chollet (2001). Combining gmms with support vector machines for text-independent speaker verification. In *euro01*, pp. 1757–1760.

Klatt, D. (1976). A digital filter bank for spectral matching. In *icassp76*, pp. 573–576.

Klusacek, D., J. Navratil, D. Reynolds, and J. Campbell (2003). Conditional pronunciation modeling in speaker detection. In *icassp03*, Volume 4, pp. 804–807.

Kohler, M. and M. Kennedy (2002). Language identification using shifted delta cepstra. In *Circuits and Systems, The 45th Midwest Symposium on*, Volume 3, pp. 69–72.

Lee, C.-H., C.-H. Lin, and B.-C. Juang (1991). A study on speaker adaptation of the parameters of continuous density hidden markov models. *39*(4), 806–812.

Leggetter, C. and P. C. Woodland (1995a). Flexible speaker adaptation using maximum likelihood linear regression.

Leggetter, C. and P. C. Woodland (1995b). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov

models. *9*(2), 171–185.

Levinson, S. (1986). Continuously variable duration hidden markov models for automatic speech recognition. *1*, 29–45.

Linguistic Data Consortium (2002). *The NIST Year 2002 Speaker Recognition Evaluation Plan*. Univ. Pennsylvania, Philadelphia, PA: Linguistic Data Consortium.

Linguistic Data Consortium (2003). *The NIST Year 2003 Speaker Recognition Evaluation Plan*. Univ. Pennsylvania, Philadelphia, PA: Linguistic Data Consortium.

Liu, L., J.-L. He, and G. Palm (1996). Signal modeling for speaker identification. In *icassp96*, Volume 2, pp. 665 – 668.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1, pp. 281–297.

Martin, A., G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki (1997). The det curve in assessment of detection task performance. pp. 1895–1898.

Martin, A. and M. Przybocki (2003). Nist's assessment of text-independent speaker recognition performance.

Meuwly, D. and A. Drygajlo (2001). Forensic speaker recognition based on a bayesian framework and gaussian mixture modelling (gmm). pp. 145–150.

Milner, B. (2002). A comparison of front-end configurations for robust

speech recognition. In *icassp02*, Volume 1, pp. 797–800.

Nadeu, C., J. Hernando, and M. Gorricho (1995). On the decorrelation of filter-bank energies in speech recognition. In *euro95*, pp. 1381–1384.

Nadeu, C., P. Pachs-Leal, and B. Juang (1997). Filtering the time sequences of spectral parameters for speech recognition. *22*, 315–332.

Nadue, C., D. Macho, and J. Hernando (2001). Time and frequency filtering of filter-bank energies for robust hmm speech recognition. *34*, 93–114.

National Institute of Standards and Technology (2003). *NIST 2003 Speaker Recognition Workshop*. College Park, MD: National Institute of Standards and Technology.

Navratil, J., Q. Jin, W. Andrews, and J. Campbell (2003). Phonetic speaker recognition using maximum-likelihood binary-decision tree models. In *icassp03*, Volume 4, pp. 796–799.

Neuberg, E. P. (1971). Markov models for phonetic text. *jasa 50*, 116(A).

Ostendorf, M., V. V. Digalakis, and O. A. Kimball (1996). From HMM's to segmental models: a unified view of stochastic modeling for speech recognition. *ieeesap 4*(5), 360–378.

Pelecanos, J. and S. Sridharan (2001). Feature warping for robust speaker verification. In *odyssey01*, pp. 213–218.

Peskin, B., J. Navratil, J. Abramson, D. Jones, D. Klusacek, D. Reynolds, and B. Xiang (2003). Using prosodic and conversational features for high-performance speaker recognition: Report from jhu ws'02. In

icassp03, Volume 4, pp. 792–795.

Poritz, A. B. (1982). Linear predictive hidden markov models and the speech signal. In *icassp82*, pp. 1291–1294.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. IEEE*, Volume 77, pp. 257–286.

Rabiner, L. R., S. E. Levinson, and M. M. Sondhi (1983). On the application of vector quantization and hidden markov models to speaker-independent, isolated work recognition. *Bell System Tech. J. / 62*(4), 1075–1105.

Reynolds, D., W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang (2003). The supersid project: Exploiting high-level information for high-accuracy speaker recognition. In *icassp03*, Volume 4, pp. 784–787.

Reynolds, D. A. (1992). *A Gaussian mixture modeling approach to text independent speaker identification*. Ph. D. thesis, Georgia Institute of Technology.

Reynolds, D. A. (1997). Comparison of background normalization methods for text-independent speaker verification. In *euro97*, pp. 963–966.

Reynolds, D. A. (2003). Channel robust speaker verification via feature mapping. In *icassp03*, Volume 2, pp. 53–56.

Reynolds, D. A. and R. C. Rose (1995). Robust text-independent speaker

identification using gaussian mixture speaker models. *ieeesap 3*, 72–83.

Reynolds, D. A., F. Q. Thomas, and B. D. Robert (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 19–41.

Richards, H. B. and J. S. Bridle (1999). The HDM: a segmental Hidden Dynamic Model of coarticulation. In *icassp99*, pp. 357–360.

Rose, R. C. and D. A. Reynolds (1990). Text-independent speaker identification using automatic acoustic segmentation. In *icassp90*, pp. 293–296.

Rosenberg, A. E., J. Delong, C.-H. Lee, B.-H. Juang, and F. K. Soong (1991). The use of cohort normalized scores for speaker verification. In *icslp92*, Volume 2, pp. 599–602.

Rosenberg, A. E., C.-H. Lee, and S. Gokcen (1991). Connected word talker verification using whole word hidden markov models. In *icassp91*, Volume 6, pp. 381–384.

Russell, M. J. (1993). A segmental HMM for speech pattern modelling. In *icassp93*, pp. 499–502.

Russell, M. J. (2005). Reducing computational load in segmental hidden markov model decoding for speech recognition. In *e-lett*, Volume 41, pp. 1408–1409.

Russell, M. J. and P. J. B. Jackson (2003). The effect of an intermediate layer on the performance of a segmental HMM. In *euro03*.

Russell, M. J. and P. J. B. Jackson (2005). A multiple-level linear/linear segmental hmm with a formant-based intermediate layer. *csl 19*, 205–

225.

Russell, M. J. and R. K. Moore (1985). Explicit modelling of state occupancy in hidden Markov models for automatpc speech recognition. In *icassp85*.

Russell, M. J. and K. M. Ponting (1990). Experiments with grand variance in the arm continuous speech recognition system. In *RSRE memorandum 4359*.

Sambur, M. R. (1975). Selection of acoustic features for speaker identification. In *IEEE Trans. on ASSP*, Volume 23, pp. 176–182.

Sankar, A. and C.-H. Lee (1996). A maximum-likelihood approach to stochastic matching for robust speech recognition. *ieeesap 4*(3), 190–202.

Shajith, M., I. H. Misra, and B. Yegnanarayana (1999). Analysis of autoassociative mapping neural networks. In *Int. Joint Conf. on Neural Networks*.

Singer, E., P. Torres-Carrasquillo, T. Gleason, W. Campbell, and D. Reynolds (2003). Acoustic, phonetic, and discriminative approaches to automatic language recognition. In *euro03*, pp. 1345–1348.

Sivian, L. J. and S. D. White (1933). On minimum audible sound fields. *jasa 5*(60), 288–321.

Smith, N. and M. Gales (2002). Using svms and discriminative models for speech recognition. In *icassp02*, Volume 1, pp. 77–80.

Sonmez, K., E. Shriberg, L. Heck, and M. Weintraub (1998). Modeling dy-

namic prosodic variation for speaker verification. In *icslp98*, Volume 7, pp. 3189–3192.

Soong, F. and A. Rosenberg (1988). On the use of instantaneous and transitional spectral information in speaker recognition. *36*, 871–879.

Soong, F. K., A. E. Rosenberg, L. R. Rabiner, and B. H. Juang (1985). A vector quantization approach to speaker recognition. In *icassp85*, pp. 387–390.

Stevens, K. (1971). Sources of inter- and intra- speaker variability in the acoustic properties of speech sounds. In *Proc. of 7th International Congress of Phonetic Sciences*, pp. 206–232.

Stevens, S. S., J. Volkman, and E. Newman (1937). A scale for the measurement of the psychological magnitude of pitch. *jasa 8*(3), 185–190.

Tappert, C. (1976). A markov model acoustic phonetic component for automatic speech recognition. In *icassp76*, Volume 1, pp. 25–28.

Tokuda, K., H. Zen, and T. Kitamura (2003). Trajectory modeling based on hmms with the explicit relationship between static and dynamic features. In *euro03*, pp. 865–868.

Torres-Carrasquillo, P., E. Singer, M. Kohler, R. Greene, D. Reynolds, and J. Deller (2002). Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In *icslp02*.

Tosi, T., H. J. Oyer, W. B. Lashbrook, C. Pedrey, and J. Nichol (1972). Experiment on voice identification. *jasa 51*, 2030–2043.

Vaseghi, S., P. Conner, and B. Milner (1993). Speech modelling using

cepstral-time feature matrices in hidden markov models. In *Communications, Speech and Vision, IEE Proceedings*, Volume 140, pp. 317–320.

Viikki, O. and K. Laurila (1998). Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication 25*, 133–147.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. pp. 260–269.

Wan, V. and W. Campbell (2000). Support vector machines for verification and identification. In *Neural Networks for Signal Processing X, Proceedings of the 2000 IEEE Signal Processing Workshop*, pp. 775–784.

Wan, V. and S. Renals (2003). Svmsvm: Support vector machine speaker verification methodology. In *icassp03*, pp. 221–224.

Wan, V. and S. Renals (2005). Speaker verification using sequence discriminant support vector machines. In *ieeesap*, Volume 13, pp. 203–210.

Weber, F., L. Manganaro, B. Peskin, and E. Shriberg (2002). Using prosodic and lexical information for speaker identification. In *icassp02*.

Wellekens, C. J. (1987). Explicit time correlation in hidden markov models for speech recognition. In *ICASSP87*, pp. 384–386.

Wiewiorka, A. and D. M. Brookes (1996). Exponential interpolation of states in a hidden Markov model. *ioa96 18*(9), 201–208.

Woodland, P., C. Leggetter, J. Odell, V. Valtchev, and S. Young (1995, May). The 1994 htk large vocabulary speech recognition system. In

icassp95, Volume 1, pp. 73–76.

Yang, X., J. Bruce Millar, and I. Macleod (1996). On the sources of inter- and intra- speaker variability in the acoustic dynamics of speech. In *icslp96*, Volume 3, pp. 1792–1795.

Young, S. (1992, March). The general use of tying in phoneme-based hmm speech recognisers. In *icassp92*, Volume 1, pp. 569–572.

Young, S. J., J. Odell, D. Ollason, V. Valtchev, and P. Woodland (1997). *The HTK Book* (v2.1 ed.). Cambridge, UK: Entropic Camb. Res. Lab.

Zheng, Y.-C. and B.-Z. Yuan (1988). Text-dependent speaker identification using circular hidden markov models. In *icassp88*, Volume 13, pp. 580–582.