The Hong Kong Polytechnic University
Department of Electronic and Information Engineering

Articulatory-Feature Based Pronunciation Modelling for High-Level Speaker Verification

Zhang Shi-Xiong

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Master of Philosophy

January 2008

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best to my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

ZHANG Shixiong (Name of student)

Abstract

Articulatory-Feature Based Pronunciation Modelling for High-Level Speaker Verification

Speaker verification is a binary classification problem whose objective is to determine whether a test utterance was produced by a client speaker. Text-independent speaker verification systems typically extract speaker-dependent features from short-term spectra of speech signals to build speaker-dependent Gaussian mixture models (GMMs). While this short-term spectral approach can achieve a reasonably good performance in controlled environment, the lack of robustness to real-world environment remains a serious problem. To improve the robustness of spectral-based systems, long-term high-level features have been investigated in recent years. Among the high-level features investigated, the use of articulatory features (AFs) for constructing conditional pronunciation models (CPMs) has been very promising. The resulting models are referred to as articulatory-feature based conditional pronunciation models, or simply AFCPMs. The drawback of AFCPMs, however, is that the pronunciation models are phoneme-dependent, meaning that they require one discrete density function for each phoneme. This dissertation demonstrates that this phoneme dependency leads to speaker models with low discriminative power, especially when the amount of training data is limited. To overcome this problem, this dissertation proposes four new techniques for articulatory-feature based pronunciation modeling.

1. *Phonetic-Class Dependent AFCPM (CD-AFCPM).* In this modeling technique, the density functions are conditioned on phonetic classes instead of phonemes. The phonetic classes are created from phonemes through three different mapping functions, which are obtained by (1) vector quantizing the discrete densities

in the phoneme-dependent universal background models, (2) using the phone properties specified in the classical phoneme tree, and (3) combination of (1) and (2).

2. *Probabilistic Weighting Scheme.* In the original CD-AFCPM, all frames are considered to be equally important during the density estimation. However, frames that have a higher probability of belonging to the phonetic class being modeled should be given a greater weight. This dissertation, therefore, proposes a weighting scheme for computing the pronunciation models such that frames with a higher probability of belonging to a particular class will have a higher contribution to the model of that class. A new scoring method that uses an SVM to combine the scores generated from the phonetic-class models is also proposed.

3. *Model Adaptation.* Speaker verification based on high-level speaker features requires long enrolment utterances to be reliable. However, in practical speaker verification, it is common to model speakers based on a limited amount of enrolment data. To alleviate this problem, this dissertation proposes a new adaptation method for creating speaker models. The method not only adapts the phoneme-dependent background model but also the phoneme-independent speaker model.

4. *Articulatory-Feature Kernels.* The log-likelihood ratio scoring method in the original AFCPM does not explicitly use the discriminative information available in the training data because the target speaker models and background models are separately trained. This dissertation proposes converting the speaker models to supervectors in high-dimensional space by stacking the discrete densities in the AFCPMs. An AF-kernel is constructed from the supervectors of target

speakers, background speakers, and claimants. Then, an SVM is discriminatively trained to classify the supervectors.

These four techniques have been evaluated on the NIST 2000 dataset. The evaluation leads to five findings:

1. Among the three mapping functions, the one that combines the classical phoneme tree and Euclidean distance between AFCPMs achieves the best performance;

2. Phonetic-classes AFCPM achieves a significantly lower error rate as compared to conventional AFCPM;

3. The weighting scheme leads to better speaker models and hence helps to improve verification performance;

4. The proposed adaptation method, which uses as much information as possible from the training data, significantly outperforms the classical MAP adaptation method; and

5. The proposed AF-kernel is complementary to the likelihood-ratio scoring method, and their fusion can improve verification performance.

ACKNOWLEDGMENTS

I would like to express sincere gratitude to various bodies from The Hong Kong polytechnic University, where I have the opportunity to study with. My major debt is to my Supervisor Dr. M. W. Mak, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skill in many areas (e.g., speech, bioinformatics, machine learning, software engineering, interaction with participants), and his assistance in writing papers and this dissertation. I have learned a lot of things from him. Without his help, this study could not be completed. I would also like to thank Prof. Helen M. Meng, who is our coauthor, for her constructive comments and suggestions to improve our papers.

Besides, I would like to express my appreciation to all the professors who have taught me during in my master study. The countless discussions with my teachers and their enthusiastic disabusing have proved to be fruitful and inspiring. I would also like to thank all members of staff of the department of Electronic and Information Engineering and the clerical staff in the General Office. They have created a creative environment for me to study in.

Finally, it is my pleasure to acknowledge the Research and Postgraduate Studies Office of The Hong Kong Polytechnic University for its generous support over the past two years.

Last but not least, I am indebted to my parents for their endless support and encouragement. Without them, this study would not have the chance to be completed.

TABLE OF CONTENTS

List of Figures **v**

List of Tables **xii**

Chapter 1: Introduction **1**

1.1 Biometric Authentication 1

1.2 Definition of Speaker Recognition 2

1.3 Speaker Recognition Modules 3

1.4 State-of-the-art Speaker Verification Systems 4

1.5 Evaluation of Speaker Verification Systems 7

1.6 Motivation of the Thesis 9

Chapter 2: High-level Speaker Verification **12**

2.1 Prosodic Feature Based Methods 12

2.2 Idiolect Based Methods 14

2.3 Pronunciation modeling 14

2.3.1 Phone N-grams and Binary Trees 14

2.3.2 Cross-stream Phone Modeling 15

2.3.3 Conditional Pronunciation Modeling 15

2.3.4 Articulatory Feature-based Conditional Pronunciation Modeling 16

Chapter 3: Phonetic-Class Articulatory Feature based Conditional Pronunciation Modeling **19**

3.1 Articulatory Feature Extraction 19
3.2 Phoneme-Dependent AFCPM 21
3.2.1 Phoneme-Dependent UBMs 21
3.2.2 Phoneme-Dependent Speaker Models 22
3.2.3 Problems of Phoneme-Dependent Speaker Models 23
3.3 Phonetic-Class Dependent AFCPM 25
3.3.1 Phoneme-to-Phonetic Class Mapping Functions 26
3.3.2 Phonetic-Class Dependent UBMs 29
3.3.3 Phonetic-Class Dependent Speaker Models 32
3.3.4 Scoring Method 35
3.4 Experiments 36
3.4.1 Speech Corpora and Features 36
3.4.2 Training Procedures 38
3.4.3 Fusion of MFCC- and AFCPM-Based Systems 38
3.5 Results and Discussion 40
3.5.1 Comparing Different Mapping Functions 40
3.5.2 Comparing CD-AFCPM and PD-AFCPM 40
3.5.3 Results on Fusing High- and Low-level Features 41
3.6 Concluding Remarks 43

Chapter 4: Probabilistic-Weighted Phonetic-Class AFCPM 45

4.1 Introduction and Motivation 45
4.2 Probabilistic Weighted Phonetic-Class Dependent AFCPM (PW-CD-AFCPM) 46
4.2.1 Mapping Function 46
4.2.2 Probabilistic Mapping Weights 46
4.2.3 Probabilistic-Weighted Phonetic-Class Dependent UBMs . . . 48

4.2.4 Probabilistic-Weighted Phonetic-Class Dependent Speaker Models . . . 49
4.2.5 SVM Scoring . . . 50
4.3 Experiments and Results . . . 54
4.3.1 Procedures . . . 54
4.3.2 Results and Discussions . . . 54

Chapter 5: New Adaptation Methods for Speaker-Model Creation in High-Level Speaker Verification 57

5.1 Introduction and Motivation . . . 57
5.2 Adaptation Methods for AFCPMs . . . 60
5.2.1 Problems of MAP Adaptation for AFCPMs . . . 61
5.2.2 New Adaptation Methods for AFCPMs . . . 64
5.3 Scoring Based on Adapted Models . . . 72
5.4 Experiments and Results . . . 73
5.4.1 Procedures . . . 73
5.4.2 Results and Discussion . . . 73

Chapter 6: Articulatory-Feature based Sequence Kernel for High-Level Speaker Verification 77

6.1 Motivation . . . 78
6.2 Phonetic-Class Dependent AFCPM Supervectors . . . 78
6.3 Feature Selection . . . 79
6.4 Articulatory Feature-Based Kernels . . . 80
6.4.1 Another Interpretation of Articulatory Feature-Based LR Scoring 80
6.4.2 Deriving Kernels from Similarity Scores . . . 86
6.4.3 Comparing AF-Kernel Scoring and LR-scoring . . . 89

6.5 Experiments and Results 91
6.5.1 Procedures 91
6.5.2 Score Fusion 91
6.5.3 Results and Discussions 91

Chapter 7: CONCLUSIONS and FUTURE WORK **95**

7.1 Conclusions 95
7.2 Future Work 96
7.2.1 Robustness Analysis of Articulatory Pronunciation Modeling for High-Level Speaker Verification 96
7.2.2 Derive the AF-Kernels from Similarity Score 97
7.2.3 Derive the AF-Kernel from Distance Metric 101

Bibliography **105**

Appendix A: Phonemes and Phonetic Classes **110**

Appendix B: Proofs of Equations **112**

Appendix C: Author's Publications **115**

LIST OF FIGURES

1.1 The training and verification phases of a typical speaker verification system. 3

1.2 The training and identification phases of a typical speaker identification system. 4

1.3 The key components of a GMM-UBM speaker verification system and its scoring process. 7

1.4 Waveforms and spectrograms of the same utterance pronounced by two speakers. 9

2.1 Figure illustrating different speakers have different ways of using their articulators to produce the same phoneme. 16

3.1 Articulatory feature-based multilayer perceptrons (AF-MLP) for the place of articulation. The MLP for the manner of articulation has a similar architecture. 20

3.2 The procedure of creating the UBMs and training the mapping function for the phonetic-class dependent AFCPM. $f^G(q) \in \{f^G_{\text{VQ}}(q), f^G_{\text{P}}(q), f^G_{\text{P+VQ}}(q)\}$, $N = 46$. 22

3.3 Phoneme-dependent AFCPM background models correspond to (a) phoneme /ah/ and (b) phoneme /ow/ based on the training utterances in NIST99. (c) to (f): Phoneme-dependent speaker models of two speakers in NIST00 adapted from (a) and (b). d represents the Euclidean distance between the models pointed to by arrows. The 60 discrete probabilities corresponding to the combinations of the 6 manner and 10 places classes are nonlinearly quantized to 256 gray levels using log scale, where white represents 0 and black represents 1. The 6 manner and 10 places classes in ascending order of the axis labels are: {Silence, Vowel, Stop, Fricative, Nasal, Approximant-Lateral} and {Silence, High, Middle, Low, Labial, Dental, Coronal, Palatal, Velar, Glottal}. 25

3.4 The verification phase of phonetic-class dependent AFCPM. $f^G(q) \in \{f^G_{\mathrm{VQ}}(q), f^G_{\mathrm{P}}(q), f^G_{\mathrm{P+VQ}}(q)\}$. 26

3.5 The procedure of training the mapping function f^G_{VQ} in Method 1. . . 27

3.6 The procedure of training the mapping function $f^G_{\mathrm{P+VQ}}$ in Method 3. . 29

3.7 The procedure of training the phonetic class AF-based speaker models. 33

3.8 Phonetic-class dependent models in which the phonemes /ah/ and /ow/ are members of the phonetic class ($c = 3$ in Table 3.3). The speaker models were obtained from the training utterances of speakers 1018 and 3823 in NIST00, using the mapping function $f^{G}_{\mathrm{P+VQ}}(q)$. d represents the Euclidean distance between the models pointed to by arrows. The 60 discrete probabilities corresponding to the combinations of the 6 manner and 10 places classes are nonlinearly quantized to 256 gray levels using log scale, where white represents 0 and black represents 1. The 6 manner and 10 places classes in ascending order of the axis labels are : {Silence, Vowel, Stop, Fricative, Nasal, Approximant-Lateral} and {Silence, High, Middle, Low, Labial, Dental, Coronal, Palatal, Velar, Glottal}. 35

3.9 Linear, polynomial, and DBNN fusion. Distribution of the score vectors from an MFCC-based GMM-UBM system and a CD-AFCPM system for the first 10% of genuine and impostor trials in NIST00. 39

3.10 DET performance of phonetic-class dependent AFCPM (CD-AFCPM), phoneme-dependent AFCPM (PD-AFCPM), GMM (fBSFT and STG were applied) with mix gender, and their fusions. 43

4.1 Training the mapping function and mapping weights. SD-VQ stands for soft-decision VQ. 46

4.2 The procedure of training the probabilistic mapping weights and mapping function. 47

4.3 The procedure of training a probabilistic-weighted CD-AFCPM. . . . 50

4.4 Probabilistic-weighted CD-AFCPM in which the phonemes /ah/ and /ow/ are members of the phonetic class. The speaker models were obtained from the training utterances of speakers 1018 and 3823 in NIST00, using the mapping function $f^{G}_{\mathrm{P+VQ}}(q)$. d represents the Euclidean distance between the models connected by arrows. 51

4.5 The verification phase of probabilistic-weighted CD-AFCPM. 52

4.6 DET performance of probabilistic-weighted phonetic-class dependent AFCPM (PW-CD-AFCPM), phoneme-dependent AFCPM (PD-AFCPM), GMM (with fBSFT and STG applied), and their fusions. All curves are based on mixed-gender scores. 56

5.1 Training of unadapted phoneme-dependent AFCPM speaker models and the data-sparseness problem they may encounter. 58

5.2 The contribution of this chapter: new adaptation methods for speaker-model creation. 59

5.3 Data-set utilization in different adaptation methods. Methods A and B only use part of the available models. Methods C and D fully utilize all of the possible models that can be obtained from training data. '*' means that the corresponding model is phoneme-independent. 60

5.4 The procedure of applying MAP adaptation (Method A) to create phoneme-dependent AFCPM speaker model. 61

5.5 *Method A*. Relationship (based on real data) between the background, unadapted, and adapted AFCPMs in classical MAP (q_1=/jh/, q_2=/uw/). The linear combination in Eq. 5.1 suggests that the adapted model will lie along the straight line passing through the unadapted model and the background model. 62

5.6 Phoneme-dependent AFCPMs correspond to phoneme /ch/ of (a) speaker 1018 from NIST00, (b) background speakers from NIST99, and (c) speaker 1042 from NIST00. (d) and (e): Phoneme-dependent speaker models of the two speakers adapted from (b) using the traditional MAP adaptation (see Method A in section 5.2.1). d and r represent the Euclidean distance and the correlation coefficient between the models pointed to by arrows. The 60 discrete probabilities corresponding to the combinations of the 6 manner and 10 place classes are nonlinearly quantized to 256 gray levels using log-scale, where white represents 0 and black represents 1. 64

5.7 The information utilized for creating an adapted speaker model using the proposed adaptation method. 66

5.8 *Method B.* Relationship between the phoneme-independent speaker model, unadapted speaker models, and adapted speaker models for speakers 1018 and 1042 (q_1=/jh/, q_2=/uw/). 67

5.9 *Method C.* Relationship between the phoneme-independent speaker model, unadapted speaker models, and adapted speaker models for speaker 1018 (q_1=/jh/, q_2=/uw/). 68

5.10 *Method D.* Relationship between the phoneme-independent speaker model, unadapted speaker models, and adapted speaker models for speaker 1018. (q_1=/jh/, q_2=/uw/ and the marker '★' represents the term inside the square brackets in Eq. 5.11.) 69

5.11 Phoneme-dependent AFCPMs ((g) and (h)) of speakers 1018 and 1042 created by Method D. (a) and (c): Unadapted speaker models. (b) Phoneme-dependent background model. (d) and (f): Phoneme-independent speaker models. (e) Phoneme-independent background model. d and r represent the Euclidean distance and the correlation coefficient between the adapted models pointed to by arrows. 71

5.12 *Method E.* Relationship between the unadapted, adapted phoneme-dependent and phoneme-independent speaker models for speaker 1018 /jh/, /uw/ and corresponding phoneme-dependent and phoneme-independent background models in Method E. 72

5.13 The distribution of all adapted phoneme-dependent speaker models and phoneme-dependent background models in principal component space for speaker 1018 and 1042 based on Method A (left) and Method D (right). 73

5.14 DET performance of AFCPM speaker verification system using different adaptation methods. 75

5.15 DET performance of AFCPM (MAP), AFCPM (Method D), GMM and their fusions. 76

6.1 The training procedure of the AF kernel-based high-level speaker verification system. 79

6.2 The procedure of extracting phonetic-class AFCPM supervectors for AF kernel-based high-level speaker verification. 80

6.3 The procedure of selecting relevant features from CD-AFCPM supervectors. Columns represent speakers and rows represent features. . . 81

6.4 Effect of feature selection on CD-AFCPM supervectors. A row with small variation (almost identical color intensity) suggests that the corresponding feature is not speaker dependent and therefore can be removed without scarifying classification accuracy. 82
6.5 An alternative implementation of the traditional log-likelihood scoring in CD-AFCPM speaker verification. 85
6.6 The effect of the normalization term $\frac{1}{\sqrt{A_b}}$ in the mapping $\varphi(\cdot)$. 89
6.7 The verification phase of an AF-kernel based speaker verification system. 90
6.8 Score fusion with and without normalization. 92
6.9 DET produced by LR scoring, AF-kernel scoring, acoustic GMM-UBM, and their fusions. 93

7.1 EERs achieved by the CD-AFCPM and PD-AFCPM at different accuracies of the manner and place MLPs. 97
7.2 Implementing the traditional log-likelihood scoring of CD-AFCPM speaker verification in KL divergence form. 102

LIST OF TABLES

2.1 The summary of high-level features in Speaker Verification (Adopted from [1]). 18

3.1 Articulatory properties and the number of classes in each property. . 21

3.2 The mapping between the phonemes and phonetic classes based on the classical phoneme tree for three different values of G. See Appendix A for the detailed relationship between the phonemes and the phonetic-classes. 28

3.3 The relationship between phonemes and phonetic classes in the mapping function $f^{G}_{\mathrm{P+VQ}}(q)$, i.e., Eq. 3.6. VQ: vector quantization; P: phoneme properties. Phonemes are firstly divided into 8 groups according to the phoneme properties (See Table A.1 in Appendix A). Then, some of these groups are further divided into subgroups via VQ. 30

3.4 A 16-frame example of an aligned phoneme, phonetic class sequences and their corresponding AF streams $\{l^{\mathrm{M}}_t$ and l^{P}_t, $t = 1, \ldots, 16\}$. The manner class labels, l^{M}_t, and place class labels, l^{P}_t, are determined by Eq. 3.1. 31

3.5 The phoneme-dependent and phonetic-class dependent AFCPMs correspond to phoneme /eh/, /ah/, /ow/ and the third phonetic class ($c = 3$). The models were obtained by using data shown in Table 3.4 and Eqs. 3.2 and 3.7. 32

3.6 The purposes of the databases used in this study. 37

3.7 EERs obtained by phoneme-dependent AFCPM (PD-AFCPM) and phonetic-class dependent AFCPM (CD-AFCPM) using three different phoneme-to-phonetic class mapping methods. "Equally weighted" means that $w_c = 1$ in Eq. 3.10 for all c. The p-values between the PD-AFCPM and all of the CD-AFCPM are less than 0.00001. 42

3.8 EERs obtained by acoustic GMM, phoneme-dependent AFCPM (PD-AFCPM) + GMM, and phonetic-class dependent AFCPM (CD-AFCPM) + GMM. The EERs corresponding to CD-AFCPM are based on class-weighted mixed-gender scenario (see Table 3.7). Note that the fusion of phonetic-class AFCPM and GMM is based on the phonetic-class AFCPM that uses the mapping function $f^G_{\text{P+VQ}}$. The p-values between PD-AFCPM+GMM and CD-AFCPM+GMM are less than 0.00001. . 44

4.1 EERs obtained by CD-AFCPM and PW-CD-AFCPM using the mapping function $f^G_{\text{P+VQ}}(q)$. The p-values between the two EERs is 0.00005. 54

4.2 EERs obtained by fusing acoustic GMM-UBM systems with PD-AFCPM, CD-AFCPM and PW-CD-AFCPM. The CD-AFCPM and PW-CD-AFCPM use the mapping function $f^G_{\text{P+VQ}}(q)$. 55

5.1 EERs obtained by phoneme-dependent AFCPMs created by MAP-based adaptation methods described in Section 5.2. The p-values between the classical MAP and the new adaptation methods are listed in the last column. 74

6.1 The EERs of AF kernel-based speaker verification systems using PD-AFCPM and CD-AFCPM supervectors without feature selection. . . 92

6.2 EER achieved by the AF kernel-based speaker verification system using CD-AFCPM supervectors with and without feature selection. 92

A.1 The relationship between phonemes and phonetic classes obtained from the classical phoneme tree [2] when the total number of phonetic classes $G = 8$. 110
A.2 The relationship between phonemes and phonetic classes obtained from the classical phoneme tree [2] when $G = 11$. 111
A.3 The relationship between phonemes and phonetic classes obtained from the classical phoneme tree [2] when $G = 13$. 111

Chapter 1

INTRODUCTION

1.1 Biometric Authentication

Security protection is critical for today's business environment and personal life. In particular, security protection has become prevalent in: (1) financial transactions, (2) access control, (3) computers and networks, and (4) personal and public safety [3]. Most commercial security systems employ artificial features for authentication, e.g., passwords, PIN, and smart ID cards. This artificial information, however, can be potentially forged and some of them could be stolen or forgotten. Better and more effective identification and authentication methods are now in great demand.

With recent technological advances in audio and visual microelectronic systems, reliable automatic authentication systems have become a commercial and practical reality. Biometric systems use automated methods to verify or recognize the identity of a person based on some physiological or behavioral characteristic (such as a fingerprint or face pattern) and/or on some aspects of behaviors (such as voice, handwriting, or keystroke patterns [3]). Since biometric systems do not identify a person by what he or she knows (a code) or possesses (a card), but by a unique characteristic that is difficult for a different individual to reproduce, the possibility of forgery is greatly reduced. However, a significant drawback of most biometrics is that they require specialized client-side measuring equipment, e.g., fingerprint readers, cameras with special lighting conditions, iris scanners, etc.

1.2 Definition of Speaker Recognition

Automatic speaker recognition [4], [5] or voice recognition is the task of determining a person's identity based on his or her own voices. Speaker recognition can be generally categorized into *speaker verification* and *speaker identification*. The former is to determine whether the voice of the claimant matches the voice of the claimed identity, whereas the latter is to identify a speaker from a set of previously enrolled speakers given an input speech utterance. Because speaker verification involves a binary comparison, the accuracy is independent of population size. On the other hand, the accuracy and response time of a speaker identification system degrades with an increasing number of registered speakers.

Speaker recognition can also be divided into text-dependent and text-independent. In text-dependent systems, the same set of keywords are used for enrollment and recognition. In text-independent systems, on the other hand, different phrases or sentences are used. Although text-dependent systems require user cooperation, they usually outperform text-independent systems because precise and reliable alignment between the unknown speech and reference templates can be made. However, text-independent systems are more appropriate for forensic and surveillance applications where pre-defined key words are not available and the users are usually not cooperative or not aware of the recognition task.

Compared with other biometric traits, speaker recognition has three distinct advantages: Firstly, speaker recognition systems do not require specialized hardware for the user interface; the only requirement is a microphone. Secondly, speech is a natural signal to produce and can be easily delivered via nowadays ubiquitous telecom systems. Finally, in some applications, speech is the main communication media (e.g., telephone-based transactions).

1.3 Speaker Recognition Modules

Speaker recognition can be divided into two distinct phases: a training phase and a recognition (verification or identification) phase. Typically, a speaker recognition system is composed of a front-end feature extractor, a number of speaker models, and a decision unit. Figure 1.1 illustrates a typical speaker verification system and Figure 1.2 illustrates a typical speaker identification system.

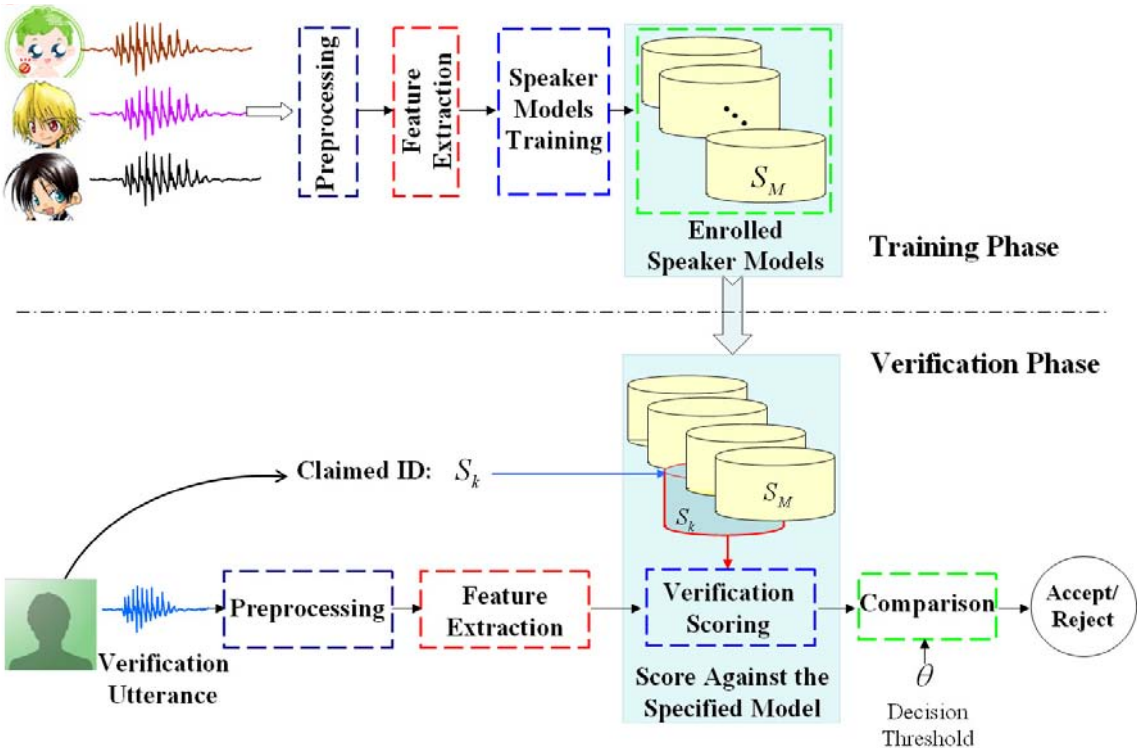


Figure 1.1: The training and verification phases of a typical speaker verification system.

The feature extractor is to derive speaker-specific features from speech signals. These features are then characterized by the speaker models. To verify a claimant, the matching score between the claimant's utterance and the model of the claimed identity is compared with a threshold, with the claimant being accepted (rejected) if the score is larger (smaller) than the threshold, as illustrate in Figure 1.1. To

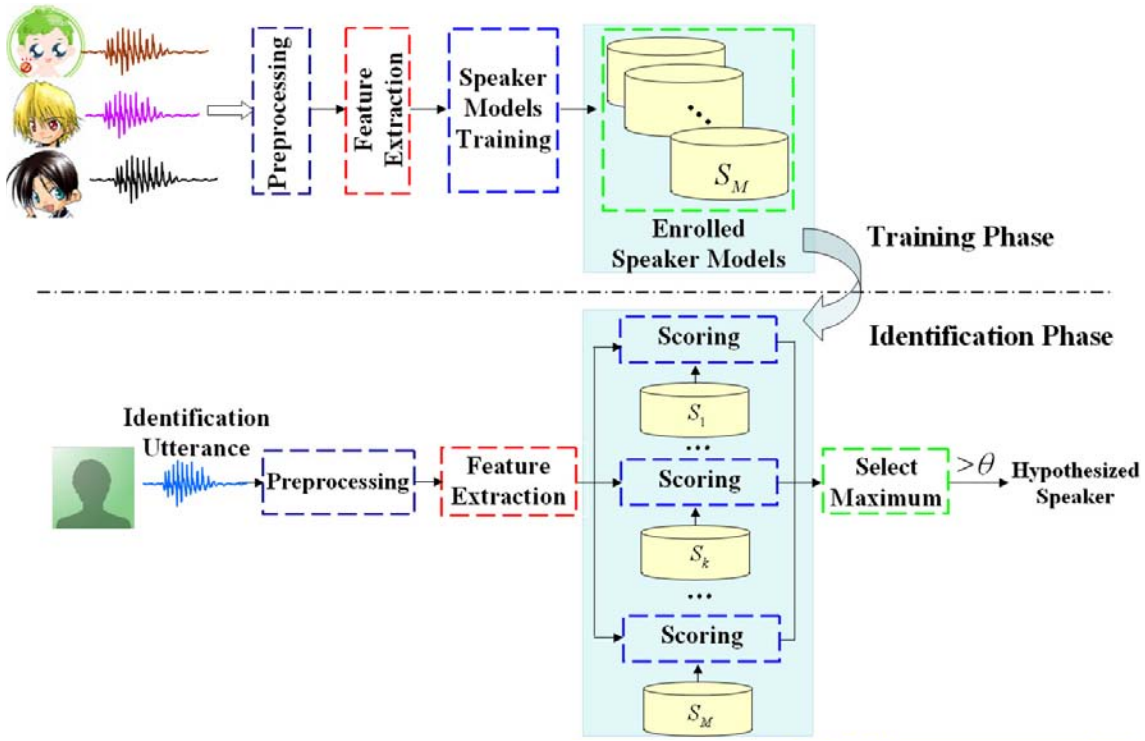


Figure 1.2: The training and identification phases of a typical speaker identification system.

identify an unknown speaker, his/her voice is compared with all speaker models. Then, the decision unit selects the closest matched model, as shown in Figure 1.2. This dissertation aims to investigate and improve feature extraction, speaker modeling, and scoring verification in speaker verification.

1.4 State-of-the-art Speaker Verification Systems

State-of-the-art text-independent speaker verification systems typically use simple but effective Gaussian mixture models (GMMs) to represent the short-term spectral characteristics of target speakers and a universal background model (UBM) to represent the spectral characteristics of a general population [6]. The Gaussian mixture model (GMM) is a density estimator and is one of the most commonly used classi-

fiers in pattern recognition. The mathematical form of an M-component GMM with D-dimensional inputs for a given speaker s is

$$\begin{aligned} p(\boldsymbol{x}|\Lambda_s) &= \sum_{i=1}^{M} w_i^s p_i^s(\boldsymbol{x}) \\ &= \sum_{i=1}^{M} w_i^s \ \frac{1}{(2\pi)^{D/2}|\Sigma_i^s|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i^s)'(\Sigma_i^s)^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i^s)\right) \end{aligned} \tag{1.1}$$

where $\sum_{i=1}^{M} w_i^s = 1$, and $w_i^s, \boldsymbol{\mu}_i^s$, and Σ_i^s are the mixture weight, mean vector, and covariance matrix of the i-th Gaussian component, respectively.

Before enrolling a client, a UBM needs to be created. Given a set of short-term spectral feature vectors, $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$, extracted from the speech of a large population, the UBM's parameters $\{w_i^b, \boldsymbol{\mu}_i^b \Sigma_i^b\}$ can be estimated by the Expectation-Maximization (EM) algorithm [7]. Then to enroll a target speaker, his/her GMM can be obtained by adapting from the UBM via the Maximum a Posteriori (MAP) formulation [6]. The details of the adaptation are as follows.

Given a UBM $\{w_i^b, \boldsymbol{\mu}_i^b \Sigma_i^b\}$ and the feature vectors $\boldsymbol{X}^s = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_{T_s}\}$ of a target speaker s, the probabilistic alignment between the feature vectors and the mixture components of the UBM is determined. That is, for mixture i in the UBM, the following posterior probability is computed:

$$\Pr(i|\boldsymbol{x}) = \frac{w_i^b p_i^b(\boldsymbol{x})}{\sum_{j=1}^{M} w_j^b p_j^b(\boldsymbol{x})}. \tag{1.2}$$

$\Pr(i|\boldsymbol{x})$ and $\boldsymbol{x}$ are then used to compute the sufficient statistics for the mixture weight,

mean vector, and covariance matrix:[1]

$$
\begin{aligned}
n_i &= \sum_{t=1}^{T} \Pr(i|\boldsymbol{x}), \\
E_i(\boldsymbol{x}) &= \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i|\boldsymbol{x})\boldsymbol{x}, \\
E_i(\boldsymbol{x}^2) &= \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i|\boldsymbol{x}^2)\boldsymbol{x}^2.
\end{aligned}
\tag{1.3}
$$

Note that Eqs 1.2 and 1.3 implement the expectation step and part of the maximization step in the EM algorithm. These new sufficient statistics are used to update the i-th componant of the UBM as follows:

$$
\begin{aligned}
\widehat{w}_i^s &= \left[\alpha_i^w n_i/T + (1-\alpha_i^w) w_i^b\right] \gamma, \\
\widehat{\boldsymbol{\mu}}_i^s &= \alpha_i^m E_i(\boldsymbol{x}) + (1-\alpha_i^m)\boldsymbol{\mu}_i^b, \\
\widehat{\sigma}_i^s &= \alpha_i^v E_i(\boldsymbol{x}^2) + (1-\alpha_i^v)\left((\sigma_i^b)^2 + (\boldsymbol{\mu}_i^b)^2\right) - (\widehat{\boldsymbol{\mu}}_i^s)^2.
\end{aligned}
\tag{1.4}
$$

Then, in the next iteration, the new adapted parameters $\{\widehat{w}_i^s, \widehat{\boldsymbol{\mu}}_i^s, \widehat{\sigma}_i^s\}$ replace the old ones $\{w_i^b, \boldsymbol{\mu}_i^b, \sigma_i^b\}$ and Eqs. 1.2 to 1.4 are repeated. The scaling factor γ is computed over all adapted mixture weights to ensure that they sum to unity. The adaptation coefficient controlling the balance between the old and new estimates is α_i^ρ, where $\rho \in \{w, m, v\}$. It is defined as follows:

$$
\alpha_i^\rho = \frac{n_i}{n_i + r^\rho} \tag{1.5}
$$

where r^ρ is a fixed relevance factor.

During verification, a likelihood-ratio (LR) test is employed to obtain a score that represents how likely the claimant is the true speaker as opposes to an impostor. The LR scoring process is illustrated in Figure 1.3.

---

[1] $\boldsymbol{x}^2$ is shorthand for $\mathrm{diag}(\boldsymbol{x}\boldsymbol{x}')$.

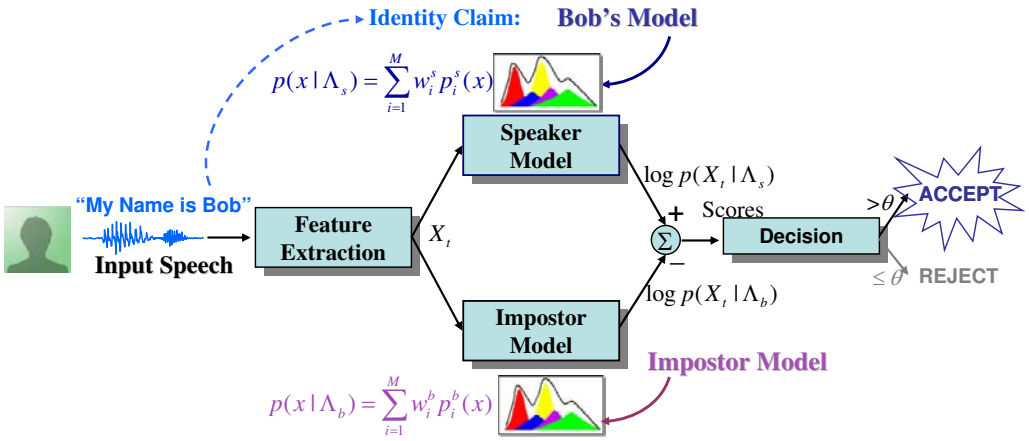


Figure 1.3: The key components of a GMM-UBM speaker verification system and its scoring process.

1.5 Evaluation of Speaker Verification Systems

As mentioned earlier, for each verification session, a verification score is compared with a decision threshold to decide whether the claimant should be accepted or should be rejected. Given a set of verification scores, the performance of a speaker verification system can be specified in terms of two types of errors:

1. *False Rejection Rate (FRR)* $P_{fr|target}$: The chance of misclassifying a true speaker as an impostor. This is also called miss probability.

2. *False Acceptance Rate (FAR)* $P_{fa|nontarget}$: The chance of falsely identifying an impostor as a true speaker. This is also called false alarm probability.

In addition to these two error rates, it is also common to report the equal error rate (EER)—the error rate at which $P_{fr|target} = P_{fa|nontarget}$. Because the false rejection rate and false acceptance rate depend on the decision threshold, a $\{P_{fr|target}, P_{fa|nontarget}\}$ pair represents one operating point of the system under evaluation. To provide more information about system performance, it is necessary to evaluate the system for a range of thresholds. With a large threshold, the system is more likely to correctly

reject impostors, but it is also more likely to falsely reject true speakers. On the other hand, with a small threshold, the system is more likely to correctly accept true speakers, but it is also more likely to falsely accept imposters. Varying the threshold from small to large results in a receiver operating characteristic (ROC) curve, which shows the tradeoff between the probability of false rejections and the probability of false alarms. The speaker verification community uses a variant of the ROC curves called detection error tradeoff (DET) curves [8], which has now become a standard metric for compaing speaker verification performance. In a DET plot, the axes' scales are normally deviated so that Gaussian distributed scores result in a straight line; unlike the ROC plots, the advantage of using DET plots is that systems with very good performance (i.e. low EER) can be compared easily.

In addition to DET curves, speaker verification systems are also compared based on the detection cost:

$$C_{det} = C_{fr} \times P_{fr|target} \times P_{target} + C_{fa} \times P_{fa|nontarget} \times P_{nontarget}$$

where C_{fr} and C_{fa} are the cost of making a false rejection error and false acceptance error, respectively, and where P_{target} and $P_{nontarget}$ are, respectively, the chance of having a true speaker and an impostor. Typical values of these figures are $C_{fr} = 10$, $C_{fa} = 1$, $P_{target} = 0.01$, and $P_{nontarget} = 0.99$ [3]. These values give an expected detection cost of approximately 1.0 for a system without any knowledge of the speakers. The operating point at which the detection cost C_{det} is at a minimum can be plotted on top of the DET curve.

Because the performance of speaker verification systems depends on the amount of training data, acoustic environment, and the length of test segments, it is very important to report this information in any performance evaluations so that performance of different systems and techniques can be compared. Thus, the NIST established a common set of evaluation data and protocols [9] in 1996. Although only focus-

ing on conversational speech, the NIST speaker recognition evaluations are one of the most important benchmark tests for speaker verification techniques. This dissertation follows the evaluation protocol of NIST2000.

1.6 Motivation of the Thesis

The main challenges in speaker verification are the following:

- *Limitation on the amount of enrollment data.*
- *Robustness to intra-speaker variability.*
- *Robustness to background noise and channel effects.*

One advantage of using short-term spectral features is that different speakers exhibit different spectral characteristics in their speech, and therefore promising results can be obtained from a limited amount of training data. As an illustrative example, the spectral patterns of two speakers uttering the same text is shown in Figure 1.4. Evidently, the spectral patterns of theses two speakers show substantial difference. However, the lack of robustness to the background noise, mismatched acoustic con-

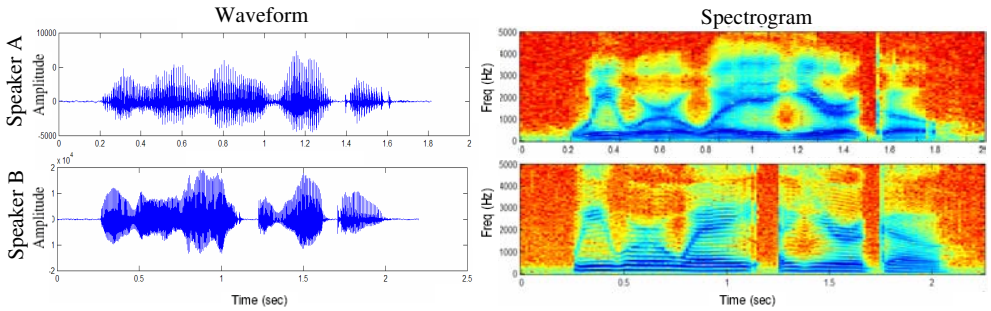


Figure 1.4: Waveforms and spectrograms of the same utterance pronounced by two speakers.

ditions, and intra-speaker variation remain a serious problem. Although approaches

such as feature transformation [10], model transformation [11], and score normalization [12] have shown promise in reducing the mismatches, these methods have almost reached their limit in terms of error rate reduction.

In order to further reduce error rate, researchers have started to investigate the possibility of using long-term, high-level speech characteristics to characterize speakers. The idea is based on the observation that humans rely not only on the low-level acoustic information but also on some high-level information to recognize each other [13]. This high-level information can be the deep bass and timber of a voice, a friend's unique laugh, or the special usage of a particular word of phrase. There is convincing evidence supporting this idea. For example, studies in speech prosody have shown that individual speakers exhibit substantial differences in voluntary speaking behaviors such as lexicon, prosody, intonation, pitch range, and pronunciation [14,15]. Studies in linguistics have shown that speaking styles (e.g., read speech vs. spontaneous speech) have significant effect on pronunciation patterns [16]. Kuehn and Moll [17] measured the velocity and displacement of tongue during speech production and found appreciable variation of these two measurements among different speakers. Shaiman et al. [18] used X-ray to capture the movement of upper lip and jaw and found substantial speaker-dependent patterns in the articulator coordination.

This work aims to investigate one of the high-level speaker features—the pronunciation characteristics—for speaker verification. Several new modeling techniques based on articulatory features are proposed for this purpose. The remainder of the dissertation is organized as follows. Some background information on high-level speaker verification is introduced in Chapter 2. Chapter 3 presents the articulatory features and explains how they can be extracted from speech signals. The chapter then outlines the phoneme-dependent pronunciation models and discusses the problems that may arise when the amount of training data is limited. To address this problem a phonetic-class dependent pronunciation modeling technique is proposed, and its advantages are demonstrated via evaluations on the NIST2000 corpus. In Chapter 4, we

extend the articulatory-feature based modeling approach to a probabilistic-weighted one. A new scoring method that uses an SVM to combine the scores generated from the phonetic-class models is also described in this chapter. Chapter 5 proposes a new adaptation method for creating speaker models under a limited amount of enrollment data.

Chapter 2

HIGH-LEVEL SPEAKER VERIFICATION

High-level speaker verification can be divided into three main categories:

- *prosodic feature based methods* that capture the patterns of sounds and rhythms of speakers,
- *idiolect based methods* that look at how individuals use a language, and
- *pronunciation modeling* that looks at how individuals pronounce a particular word or phoneme.

The state-of-the-art high-level features in speaker verification and their modeling methods are summarized in Table 2.1.

2.1 Prosodic Feature Based Methods

Our daily experience suggests that different speakers produce speech at different rhythms. In fact, studies in speech prosody have shown that individual speakers exhibit substantial differences in intonation and pitch range [14, 15]. In the context of automatic speaker recognition, prosodic features include (1) word, phone, and segmental durations, (2) pause durations and frequency, (3) pitch-related information, and (4) duration of turn-taking in conversational speech. Peskin et al. [19] provide a comprehensive list of prosodic features and the ways of extracting them from speech.

Previous research in speaker recognition has shown that prosodic information can be extracted automatically to enhance the robustness of speaker recognition systems [20–24]. There are two main ways of applying prosody information to speaker

recognition. In the first approach, global statistics (e.g., mean and standard deviation) of prosodic features obtained from enrollment and verification utterances are compared [23]. In the SuperSID project [25], Adami et al. [20] built a baseline system that extracts the global distribution of pitch and energy values. The system is similar to the classical spectral-based ones [6], but using energy- and pitch-based features instead of spectral features. Carey et al. [23] showed that prosodic features can be appended to spectral vectors for statistical modeling. One potential problem with this global statistics approach is that it fails to capture information about local variation in the speaking rhythm. Although this problem can be addressed in part by using the long-term statistics of the features' time derivatives, as in the baseline system of [20], the time scale and complexity of pitch variation are far more complex than what the temporal derivative can capture.

The second approach aims to alleviate the limitations of the first one. Instead of estimating the statistics of prosody features, this approach focuses on representing and comparing the temporal trajectories of the prosodic contours, e.g., by applying dynamic time warping (DTW) to compare the pitch contours between two utterances of the same text [26,27]. In another example, to handle the time scale and complexity of pitch variation, Sonmez et al. [21] used a linear piecewise model to fit the pitch contour followed by statistical modeling of the parameters of the piecewise model. This type of approach has the advantage of being able to capture the speaker-specific temporal dynamic events, but they generally require comparison of the same spoken text [20]. To relax this requirement, Andre et al. [20] proposed using bigrams to model the prosodic dynamics of the fundamental frequency and energy trajectories. Instead of using the numerical values of pitch and energy, a sequence of symbols describing the pitch and energy slope states (rising and falling), segment duration, and phone or word context are used to train an n-gram prosodic classifier. Results show that modeling the pitch- and energy-contour dynamics outperforms modeling the global distributions of pitch and energy values. Another advantage is that the

method is more robust to errors in the pitch and energy estimation, because all the error-prone numerical values have been quantized to symbols. The sparsity of the prosodic features, however, means that a considerable amount of training data is required.

2.2 Idiolect Based Methods

This category of approaches is based on the notion that different speakers may use a language differently to express the same meaning. In particular, some speakers may use a particular word or phrase more often than others. This suggests that it is possible to train speaker-dependent language models for speaker recognition. Doddington [28] used word unigrams and bigrams in the conventional likelihood ratio framework and obtained very promising results. The work has in fact led to extensive investigations on high-level features in the SuperSID project [25]. More recently, in [29], the unigrams, bigram, and trigrams of frequently occurred words were assembled into a feature vector, which was then classified by linear SVMs.

2.3 Pronunciation modeling

Because of the differences in education background, accents, and so on, different persons have different ways of pronouncing the same word. Therefore, the pronunciation patterns of individuals can be used as features for discriminating speakers.

2.3.1 Phone N-grams and Binary Trees

This category of approaches has been studied by various groups [30–32]. For example, Andrews et al. [30] use n-grams to model the phone streams obtained from a bank of open-loop language-dependent phone recognizers. Given a test utterance, each of the phone recognizers processes the utterance to produce a phone sequence. Then, the test phone sequence is compared to the phone model of the target speaker and

a universal background phonetic model to compute a likelihood ratio score. Finally, the scores from different phone streams are combined to form a single weighted score. Although the results in [30] are quite promising, the grid structure of n-gram models requires high model order for accurate modeling, which means that a large amount of training data is required for each speaker. To address this problem, Navratil et al. [32] used a binary tree model to represent the phone sequences. The tree model's flexible structure allows the statistical dependency within the long-term context of the phoneme sequences to be exploited without exponentially increase in model complexity. To deal with limited training data and robustness issues, an adaptation step and a recursive smoothing technique were applied to create the tree models.

2.3.2 Cross-stream Phone Modeling

If an utterance is recognized (tokenized) by a bank of language-dependent recognizer, the resulting phoneme sequences should theoretically exhibit some dependencies across multiple languages at a given time instance. Jin et al. [33] assumed that these token dependencies are related to how speakers articulate phonemes. After aligning the language-dependent phone streams, the speaker phonetic model and universal background phonetic model can be build via n-grams or binary-tree in the cross-stream dimension through which log-likelihood ratio scores can be computed. Results show that phone dependencies in the cross-stream and time dimensions do contain complementary information.

2.3.3 Conditional Pronunciation Modeling

Among all high-level features investigated in the SuperSID project [25], the conditional pronunciation modeling (CPM) technique [34] that extracts multilingual phone sequences from utterances achieves the best performance. CPM aims to model speaker-specific pronunciations by learning the relationship between what has been said (phonemes) and how speech is pronounced (phones). The rationale behind using

CPM for speaker verification is that different speakers have different ways of pronouncing the same phonemes. One limitation of CPM, however, is that it requires multi-lingual corpora to build speaker and background models.

2.3.4 Articulatory Feature-based Conditional Pronunciation Modeling

To overcome the limitation of CPM, Leung et al. [35] proposed using articulatory feature (AF) streams to construct CPM and called the resulting models AFCPM. AFs are abstract classes describing the movement or positions of different articulators during speech production. The idea hinges upon the linkage between the states of articulation during speech production and the actual phones produced by speakers. Because different persons have different ways of using their articulators to pronounce the same phonemes (see Figure 2.1 for an illustrative example), the articulatory patterns of individuals can be used as features for discriminating speakers. In contrast

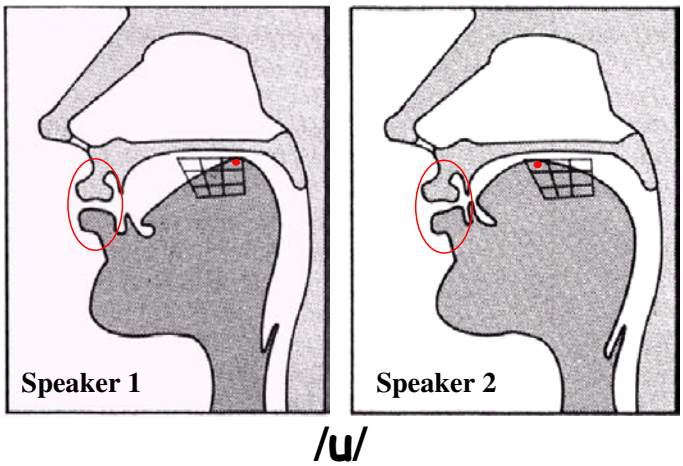
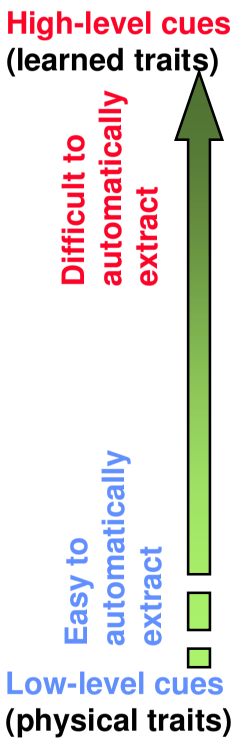


Figure 2.1: Figure illustrating different speakers have different ways of using their articulators to produce the same phoneme.

to the conventional speaker recognition systems in which short-term spectral characteristics are represented by Gaussian mixture models (GMM) [6], AFCPM-based speaker verification systems use discrete probabilistic models to represent two artic-

ulatory properties: manner and place of articulation. More specifically, the speaker models are composed of conditional probabilities of articulatory classes in these two properties, and each speaker has N phoneme-dependent discrete probabilistic models, one for each phoneme. It was found in [35] that AFCPM achieves significantly lower error rate as compared to the conventional CPM.

While promising results have been obtained, AFCPM requires a large amount of speech data for training the phoneme-dependent speaker models. Insufficient enrollment data will lead to imprecise speaker models and poor performance. To improve the accuracy of articulatory feature-based models, this dissertation proposes using phonetic-class based AFCPM. In this method, phonemes with similar manner and place of articulation are grouped together based on the similarity between the AFCPM universal background models. Then, a discrete density function is computed for each phoneme class. It was found that this phonetic-class AFCPM approach can reduce the side effect caused by the error in the phoneme recognizer and effectively solve the data sparseness problem encountered in conventional AFCPM. Experimental results show that the proposed modification leads to a significantly lower error rate as compared to the conventional AFCPM.



Feature Category	Feature Description	Feature Extractor	Feature Time Span	System Models
Pronunciations (Place of birth, education, socio-economic status, etc.)	Multilingual phone streams	Language dependent Phone-ASR	Several frames	N-gram [30]
				Binary tree [32]
	Multilingual phone cross-streams	Language dependent Phone-ASR	Several frames	N-gram [33]
				CPM [34]
	Articulatory Features	MLP & Phone-ASR	Several frames	AFCPM [35]
Idiolect (Education, socio-economic status, etc.)	Word streams	Word-ASR	Several frames	N-gram [28]
				SVM [29]
Prosodic or Rhythm (Personality type, parental influence, etc.)	F0 & Energy distribution	F0 & Energy Estimator	frame	GMM [20]
	Pitch contour	F0 Estimator & word-ASR	Several frames	DTW [26, 27]
	F0 & Energy contour & duration dynamic	F0 & Energy Estimator & Phone-ASR	Several frames	N-gram [20]
	Prosodic Statistics from F0 & duration	F0 & Energy Estimator & Word-ASR	Several frames	KNN [19]
Acoustic (Physical structure of vocal apparatus)	MFCC and its Derivatives streams	MFCC Extractor	One/several frame	GMM [6]

Table 2.1: The summary of high-level features in Speaker Verification (Adopted from [1]).

Chapter 3

PHONETIC-CLASS ARTICULATORY FEATURE BASED CONDITIONAL PRONUNCIATION MODELING

3.1 Articulatory Feature Extraction

Articulatory features (AFs) are the representations of some important phonological properties appeared during speech production. More precisely, AFs are abstract classes describing the movements or positions of different articulators during speech production. Since AFs are closely related to the speech production process, they are suitable for capturing the pronunciation characteristics of speakers.

In Leung et al. [35], the manner and place of articulation, as shown in Table 3.1, were used for pronunciation modeling. These properties describe the way and location that the air-stream along the vocal tract is constricted by the articulators. Leung et al. adopted the AF extraction approach outlined in [36]. Specifically, the AFs were automatically determined from speech signals using AF-based multilayer perceptrons (MLPs) [37] as shown in Figure 3.1. For each articulatory property, an AF-MLP takes 9 consecutive frames of 26-dimensional normalized MFCCs X_t (with consecutive frame indexes ranging from $t-4$ to $t+4$) as input to determine the posterior probabilities of the output classes at frame t. For example, given X_t at frame t, the manner MLP determines six posterior probabilities of the output classes, i.e., $P(L^{\mathrm{P}} = p|X_t)$ where $p \in \mathcal{P}$ with $\mathcal{P}$ defined in Table 3.1. Using these probabilities, the manner class label

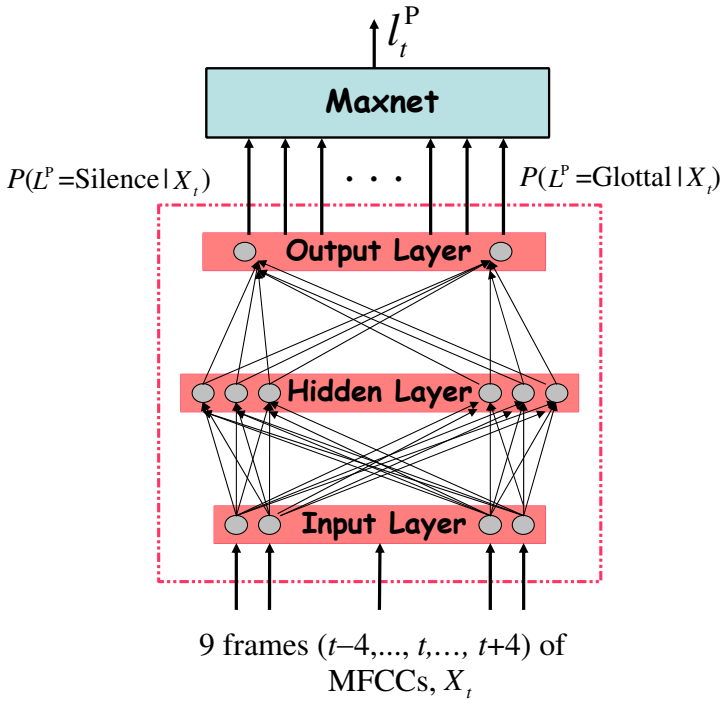


Figure 3.1: Articulatory feature-based multilayer perceptrons (AF-MLP) for the place of articulation. The MLP for the manner of articulation has a similar architecture.

$l_t^{\mathrm{M}} \in \mathcal{M}$ and place class label $l_t^{\mathrm{P}} \in \mathcal{P}$ at frame t are determined by

$$\begin{aligned} l_t^{\mathrm{M}} &= \arg\max_{m \in \mathcal{M}} P(L^{\mathrm{M}} = m | X_t) \\ l_t^{\mathrm{P}} &= \arg\max_{p \in \mathcal{P}} P(L^{\mathrm{P}} = p | X_t). \end{aligned} \tag{3.1}$$

The two AF streams—one from the manner MLP and another from the place MLP—for creating the conditional pronunciation models are formed by concatenating l_t^{M}'s and l_t^{P}'s for $t = 1, \ldots, T$, where T is the total number of frames in the utterance.

Interestingly, the AF-MLPs do not need to be very accurate for the purpose of capturing articulatory features.[1] This is mainly because their main purpose is to capture the articulatory features of speakers instead of classifying the articulatory

[1]In our experiments, the manner and place MLPs achieve an average accuracy of 79.49% and 67.69% on the HTIMIT corpus.

Articulatory Properties	**Classes**	**Number of Classes**
Manner($\mathcal{M}$)	Silence, Vowel, Stop, Fricative, Nasal, Approximant-Lateral	6
Place($\mathcal{P}$)	Silence, High, Middle, Low, Labial, Dental, Coronal, Palatal, Velar, Glottal	10

Table 3.1: Articulatory properties and the number of classes in each property.

properties. Therefore, as long as the patterns of mistakes made by these MLPs are consistent for the same speaker and different for different speakers, they can still provide valuable speaker information for building the pronunciation models. This conjecture is supported by the experimental results shown in Section 7.2.1.

3.2 Phoneme-Dependent AFCPM

3.2.1 Phoneme-Dependent UBMs

As illustrated in the left portion of Figure 3.2, N phoneme-dependent universal background models (UBMs) are trained from the AF and phoneme streams of a large number of speakers to represent the speaker-independent pronunciation characteristics. Each UBM comprises the joint probabilities of the manner and place classes conditioned on a phoneme. The training procedure begins with aligning two AF streams obtained from the AF-MLPs and a phoneme sequence obtained from a null-grammar recognizer [35]. The joint probabilities corresponding to a particular phoneme q is given by

$$\begin{aligned} & P_b^{\mathrm{PD}}(m,p|q) \\ &= P_b^{\mathrm{PD}}(L^{\mathrm{M}}=m, L^{\mathrm{P}}=p|\mathrm{Phoneme}=q, \mathrm{Background}) \\ &= \frac{\#((m,p,q) \ \text{ in the utterances of all background speakers})}{\#((*,*,q) \ \text{ in the utterances of all background speakers})} \end{aligned} \tag{3.2}$$

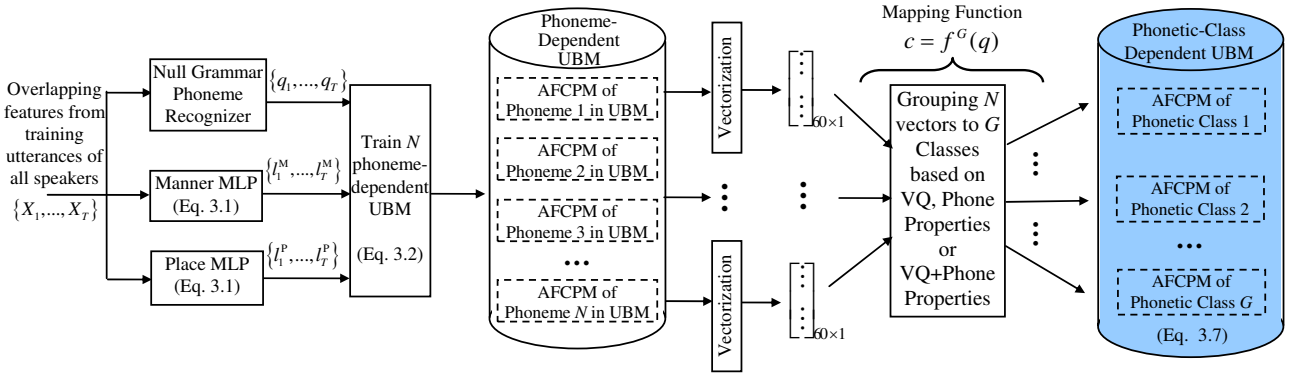


Figure 3.2: The procedure of creating the UBMs and training the mapping function for the phonetic-class dependent AFCPM. $f^G(q) \in \{f^G_{\text{VQ}}(q), f^G_{\text{P}}(q), f^G_{\text{P+VQ}}(q)\}$, $N = 46$.

where $m \in \mathcal{M}, p \in \mathcal{P}, (m, p, q)$ denotes the condition for which $L^{\text{M}} = m, L^{\text{P}} = p$, and Phoneme $= q$, $*$ represents all possible members in that class, and $\#()$ represents the total number of frames with phoneme labels and AF labels fulfilling the description inside the parentheses. For each phoneme, a total of 60 probabilities can be obtained. These probabilities are the products of 6 manner classes and 10 place classes. Therefore, a system with N phonemes has $60N$ probabilities in the UBMs. Eq. 3.2 will be used in Section 3.3.1 to train a mapping function that maps phonemes to phonetic classes.

3.2.2 Phoneme-Dependent Speaker Models

A speaker model can be obtained from speaker-dependent data as follows:

$$\begin{aligned} & P_s^{\text{PD}}(m, p \mid q) \\ &= P_s^{\text{PD}}(L^{\text{M}} = m, L^{\text{P}} = p \mid \text{Phoneme} = q, \text{Speaker} = s) \\ &= \frac{\#((m, p, q) \text{ in the utterances of speaker } s)}{\#((*, *, q) \text{ in the utterances of speaker } s)}. \end{aligned} \tag{3.3}$$

However, the accuracy of speaker models obtained by Eq. 3.3 is limited by the amount of training data available. For some phonemes (e.g., /th/, /sh/, and /v/), the number

of occurrences is too small for an accurate estimation of the joint probabilities. To overcome this data-sparseness problem, speaker models can be adapted from the UBMs. Specifically, given the background model corresponding to phoneme q, the joint probabilities $\widehat{P}_s^{\mathrm{PD}}(m, p \mid q)$ for speaker s are given by

$$\widehat{P}_s^{\mathrm{PD}}(m, p \mid q) \;=\; \beta_q P_s^{\mathrm{PD}}(m, p \mid q) + (1 - \beta_q) P_b^{\mathrm{PD}}(m, p \mid q)$$

where $\beta_q \in [0, 1]$ is a phoneme-dependent adaptation coefficient controlling the contribution of the unadapted speaker model (Eq. 3.3) and the background model (Eq. 3.2) on the adapted model. Similar to MAP adaptation of GMM-based systems [6], β_q can be obtained by

$$\beta_q = \frac{\#((*, *, q) \text{ in the utterances of speaker } s)}{\#((*, *, q) \text{ in the utterances of speaker } s) + r},$$

where r is a fixed relevance factor common to all phonemes and speakers. The purpose of r is to control the dependence of the adapted model on speaker's data. If the number of occurrences of $(*, *, q)$ is significantly smaller than r, then β_q will be very close to 0 and the estimation of the new model is less dependent on speaker's data. On the contrary, if the number of occurrences of $(*, *, q)$ is significantly greater than r, then β_q will be very close to 1 and the adapted model will become more dependent on speaker's data.

3.2.3 Problems of Phoneme-Dependent Speaker Models

While promising results have been obtained, AFCPM requires a large amount of speech data for training the phoneme-dependent speaker models. Insufficient enrollment data will lead to inaccurate speaker models and poor performance. Moreover, because the method is phoneme based, it builds phoneme-dependent models regardless of the fact that some phonemes are very similar in terms of articulatory properties.

This causes some of the background models to be almost identical. Worse yet, because the speaker models are adapted from the background models, for those "similar" phonemes that rarely occur in the speakers' utterances, the corresponding speakers models will be almost identical to the background models, making the speaker models fail to discriminate the speakers. This situation is exemplified in Figure 3.3 where the density functions of background and speaker models are illustrated as 2-D images. Evidently, there is substantial similarity between the two background models (Figures 3.3(a) and 3.3(b)). Comparisons between Figures 3.3(c) and 3.3(d) and between Figures 3.3(e) and 3.3(f) also reveal that the models of speaker 1018 are very similar to those of speaker 3823.

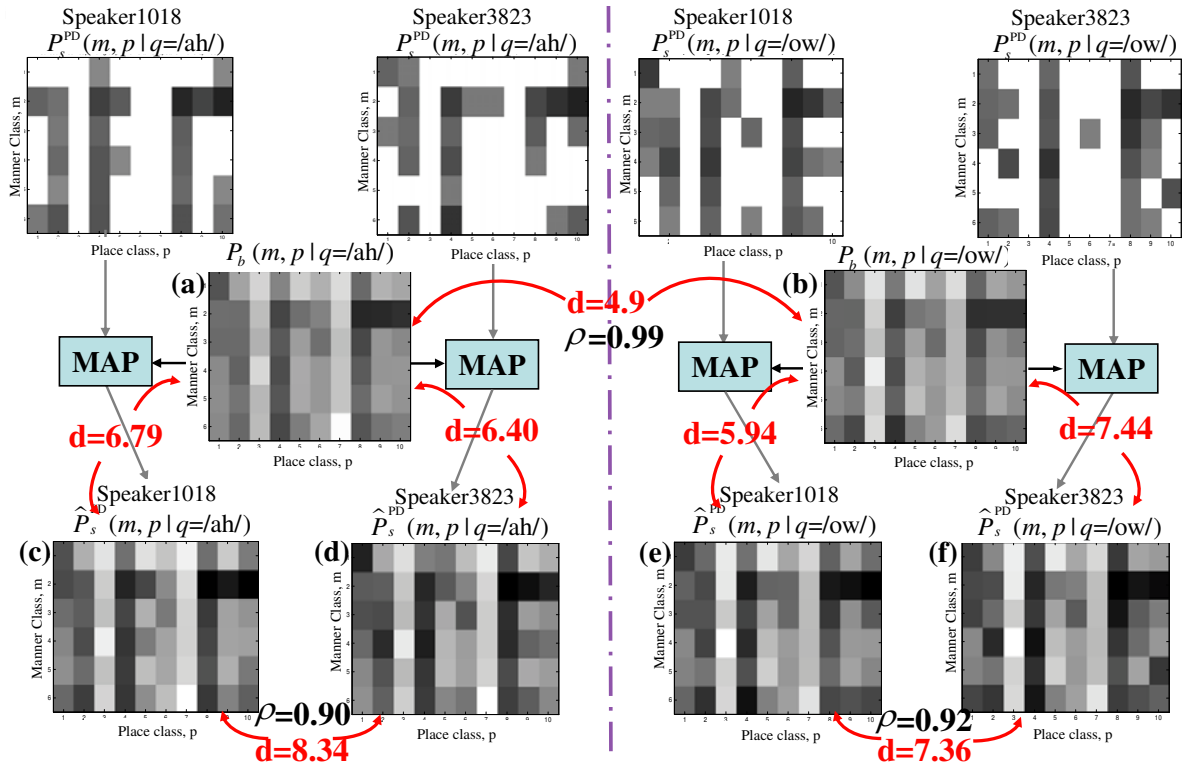


Figure 3.3: Phoneme-dependent AFCPM background models correspond to (a) phoneme /ah/ and (b) phoneme /ow/ based on the training utterances in NIST99. (c) to (f): Phoneme-dependent speaker models of two speakers in NIST00 adapted from (a) and (b). d represents the Euclidean distance between the models pointed to by arrows. The 60 discrete probabilities corresponding to the combinations of the 6 manner and 10 places classes are nonlinearly quantized to 256 gray levels using log scale, where white represents 0 and black represents 1. The 6 manner and 10 places classes in ascending order of the axis labels are: {Silence, Vowel, Stop, Fricative, Nasal, Approximant-Lateral} and {Silence, High, Middle, Low, Labial, Dental, Coronal, Palatal, Velar, Glottal}.

3.3 Phonetic-Class Dependent AFCPM

In phoneme-dependent AFCPM [35], each speaker is modeled by 60 joint probability functions of the manner and place classes conditioned on a phoneme. We found that the AFCPMs of some phonemes are very similar (e.g., see Figure 3.3), it is possible to improve the accuracy of the models by grouping the similar AFCPMs into a model set. In other words, each density function can be conditioned on a phonetic-class instead

of a single phoneme. Figures 3.2 and 3.4 illustrates the training and verification procedures of the phonetic-class dependant AFCPM, respectively.

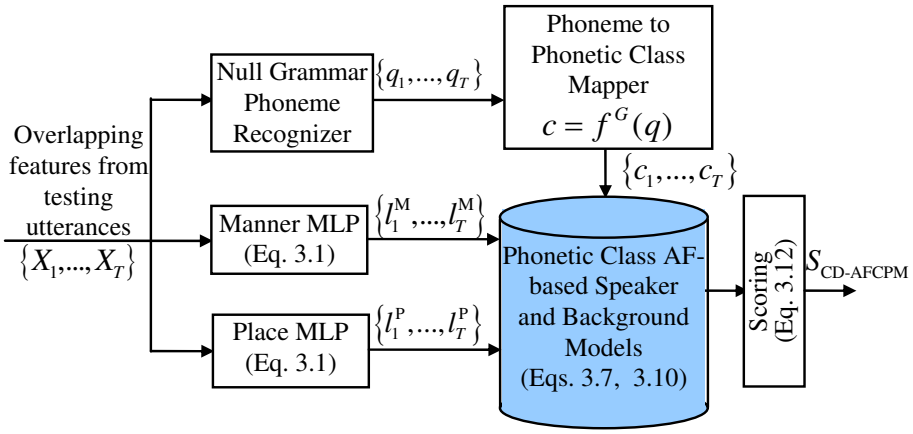


Figure 3.4: The verification phase of phonetic-class dependent AFCPM. $f^G(q) \in \{f^G_{\text{VQ}}(q), f^G_{\text{P}}(q), f^G_{\text{P+VQ}}(q)\}$.

3.3.1 Phoneme-to-Phonetic Class Mapping Functions

There are several ways of grouping phonemes: (1) according to the similarity (Euclidean distance) between the AFCPMs, (2) according to the phoneme properties as depicted in the classical phoneme tree [2], and (3) combination of (1) and (2).

Method 1: Grouping based on Euclidean distance

The phoneme-dependent UBMs, $P_b^{\text{PD}}(m, p|q)$, are vectorized to N 60-dimension vectors called AFCPM vectors (see Figure 3.2):

$$\mathbf{a}_q = \begin{bmatrix} P_b^{\text{PD}}(L^{\text{M}} = \text{`Vowel'}, L^{\text{P}} = \text{`High'}|\text{Phoneme} = q) \\ P_b^{\text{PD}}(L^{\text{M}} = \text{`Vowel'}, L^{\text{P}} = \text{`Low'}|\text{Phoneme} = q) \\ \cdots \\ P_b^{\text{PD}}(L^{\text{M}} = \text{`Lateral'}, L^{\text{P}} = \text{`Glottal'}|\text{Phoneme} = q) \end{bmatrix}$$

where $q \in \{\text{Phoneme } 1, \ldots, \text{Phoneme } N\}$. Then in this phoneme-dependent model space, K-means clustering or VQ can be applied to cluster the N AFCPM vectors into G classes. Finally, the mapping from a specific phoneme to its corresponding phonetic class index c is defined as a mapping function:

$$c = f_{\text{VQ}}^{G}(q), \quad c \in \{1, 2, \ldots, G\}. \tag{3.4}$$

The procedure of training the mapping function in Method 1 is shown in Figure 3.5.

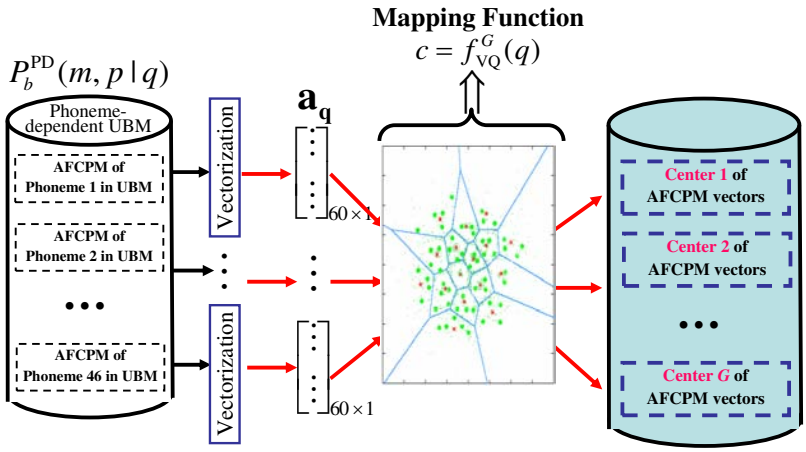


Figure 3.5: The procedure of training the mapping function f_{VQ}^{G} in Method 1.

The mapping function will be used to train the phonetic-class UBMs and speaker models, which is to be detailed in Sections 3.3.2 and 3.3.3.

Method 2: Grouping based on phoneme properties

Because the phoneme grouping in classical phoneme tree [2] is partly based on articulatory properties, we can also use the tree to determine the mapping between phonemes and phonetic classes. This results in the mapping function

$$c = f_{\text{P}}^{G}(q), \quad c \in \{1, 2, \ldots, G\}. \tag{3.5}$$

Table 3.2 shows the mapping between the phonemes and phonetic classes obtained from the classical phoneme tree [2] for three different values of G.

Phoneme q	Class label for phoneme q		
	G=8	G=11	G=13
Front Vowels: iy, ih, ey, eh, ae	1	1	1
Mid Vowels: er, ax, ah		2	2
Back Vowels: uw, uh, ow, ao, aa		3	3
Voiced Fricatives: v, dh, z, zh	2	4	4
Unvoiced Fricatives: f, th, s, sh		5	5
Whisper: hh	3	6	6
Affricates: jh, ch	4	7	7
Diphthongs: ay, aw, oy	5	8	8
Liquids: r, l, el	6	9	9
Glides: w, y			10
Voiced Consonants: b, d, g	7	10	11
Unvoiced Consonants: p, t, k			12
Nasals: m, en, n, ng	8	11	13

Table 3.2: The mapping between the phonemes and phonetic classes based on the classical phoneme tree for three different values of G. See Appendix A for the detailed relationship between the phonemes and the phonetic-classes.

Method 3: Grouping based on Euclidean distance and phoneme properties

Note that Method 1 and Method 2 group phonemes according to different criteria. Specifically, the former is based on the articulatory properties, whereas the latter is based on continuant/noncontinuant properties of phonemes. For example, phonemes are grouped in part by the vertical positions (high, middle, and low) of the tongue via the place of articulation in Method 1, whereas they are grouped by the horizontal tongue positions (front, central, and back) in Method 2. Because these two ways of phoneme characterization may complement each other, we propose a hybrid method based on the classical phoneme tree and Euclidean distance between AFCPMs to

build the third mapping function:

$$c = f^{G}_{\mathrm{P+VQ}}(q), \quad c \in \{1, 2, \ldots, G\}. \tag{3.6}$$

In this method, phonemes are grouped firstly by using phoneme properties. The phonemes in the same group are then further divided into subgroups by VQ. For example, all phonemes belonging to 'Vowels' in Table 3.2 are grouped together and then divided into 3 subgroups by using VQ. For the classes with very small number of phonemes, such as the phonetic class 'Affricates', or for those with insufficient frames for clustering, such as the phonetic class 'Liquids', their models are copied directly from phonetic-class dependent UBMs to build the mapping function. The procedure of training the mapping function using Method 3 is shown in Figure 3.6. Table 3.3 shows the mapping function $f^{G}_{\mathrm{P+VQ}}(q)$ used in this work.

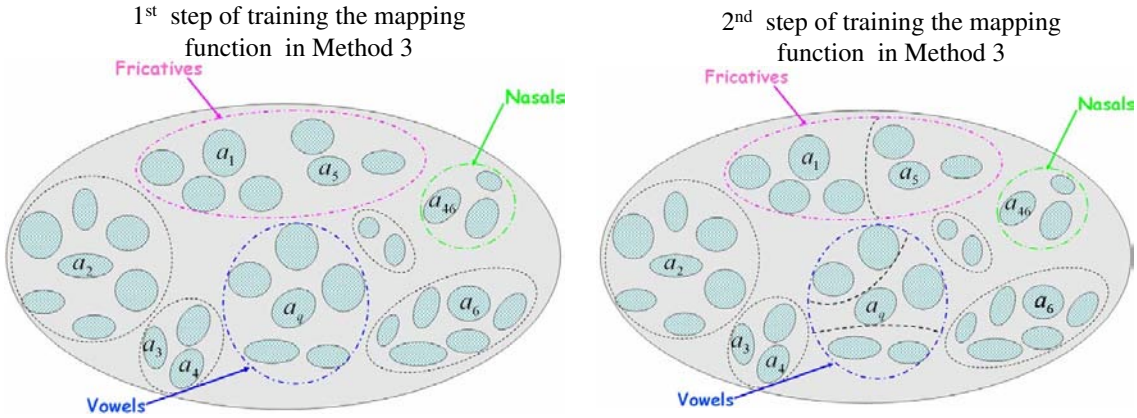


Figure 3.6: The procedure of training the mapping function $f^{G}_{\mathrm{P+VQ}}$ in Method 3.

3.3.2 Phonetic-Class Dependent UBMs

Given the mapping functions and the phoneme-dependent UBMs, phonetic-class dependent UBMs can be obtained as follows. For a particular phonetic class c, the joint

Phonetic Class c	**Phoneme** q	**Obtained by**
1	iy, uw, ih	P+VQ
2	er, uh, ax, ey	P+VQ
3	eh, ah, ow, ae, ao, aa	P+VQ
4	v, f, th, dh	P+VQ
5	z, zh, s, sh	P+VQ
6	hh	P
7	jh, ch	P
8	ay, aw, oy	P
9	r, l, el, w, y	P
10	b, d, p, t	P+VQ
11	g, k	P+VQ
12	m, en, n, ng	P

Table 3.3: The relationship between phonemes and phonetic classes in the mapping function $f^G_{\text{P+VQ}}(q)$, i.e., Eq. 3.6. VQ: vector quantization; P: phoneme properties. Phonemes are firstly divided into 8 groups according to the phoneme properties (See Table A.1 in Appendix A). Then, some of these groups are further divided into subgroups via VQ.

probabilities of the phonetic-class dependent UBMs are determined by:[2]

$$\begin{aligned} & P_b^{\text{CD}}(m,p|c) \\ &= P_b^{\text{CD}}(L^{\text{M}}=m, L^{\text{P}}=p|\text{PhoneClass}=c, \text{Background}) \\ &= \frac{\#((m,p,c)\text{in the untterances of all background speakers})}{\#((*,*,c)\text{in the untterances of all background speakers})} \end{aligned} \tag{3.7}$$

where $m \in \mathcal{M}, p \in \mathcal{P}$, and (m,p,c) denotes the condition for which $L^{\text{M}}=m, L^{\text{P}}=p$, and PhoneClass $= c$. Examples of training the phoneme-dependent and phonetic-class dependent models using the data in Table 3.4 (assuming the probabilities of unseen AF combinations are zero) are illustrated in Table 3.5.

Note that the accuracy of the mapping functions and hence the phonetic-class de-

[2]Note that $P_b^{\text{CD}}(m,p|c) \neq \frac{1}{N}\sum_{q:f^G(q)=c} P_b^{\text{PD}}(m,p|q)$.

Frame, t	Phoneme, q_t	Phonetic Class, c_t	l_t^{M}	l_t^{P}
1	/t/	c=10	Vowel	Low
2	/t/	c=10	Silence	Silence
3	/t/	c=10	Silence	Silence
4	/t/	c=10	Stop	Coronal
5	/ah/	c=3	Silence	Silence
6	/ah/	c=3	Vowel	velar
7	/ah/	c=3	Vowel	velar
8	/ah/	c=3	Vowel	velar
9	/ah/	c=3	Fricative	Low
10	/eh/	c=3	Fricative	Low
11	/eh/	c=3	Vowel	velar
12	/eh/	c=3	Vowel	velar
13	/d/	c=10	Vowel	Low
14	/ow/	c=3	Silence	Silence
15	/ow/	c=3	Fricative	Low
16	/ow/	c=3	Lateral	Low

Table 3.4: A 16-frame example of an aligned phoneme, phonetic class sequences and their corresponding AF streams $\{l_t^{\mathrm{M}}$ and $l_t^{\mathrm{P}}, t = 1, \ldots, 16\}$. The manner class labels, l_t^{M}, and place class labels, l_t^{P}, are determined by Eq. 3.1.

pendent UBMs depends on the amount of data in individual phonetic classes. Therefore, it is necessary to weight the models' density functions according to the amount of data available for training the mapping functions. Here, we propose to compute the weighting coefficients as follows:

$$w_c = \frac{\frac{\#\big((*,*,c) \text{ in the untterances of all background speakers}\big)}{\#\big((*,*,c) \text{ in the untterances of all background speakers}\big) + r_w}}{\sum_{c'=1}^{G} \frac{\#\left((*,*,c') \text{ in the untterances of all background speakers}\right)}{\#\left((*,*,c') \text{ in the untterances of all background speakers}\right) + r_w}} \tag{3.8}$$

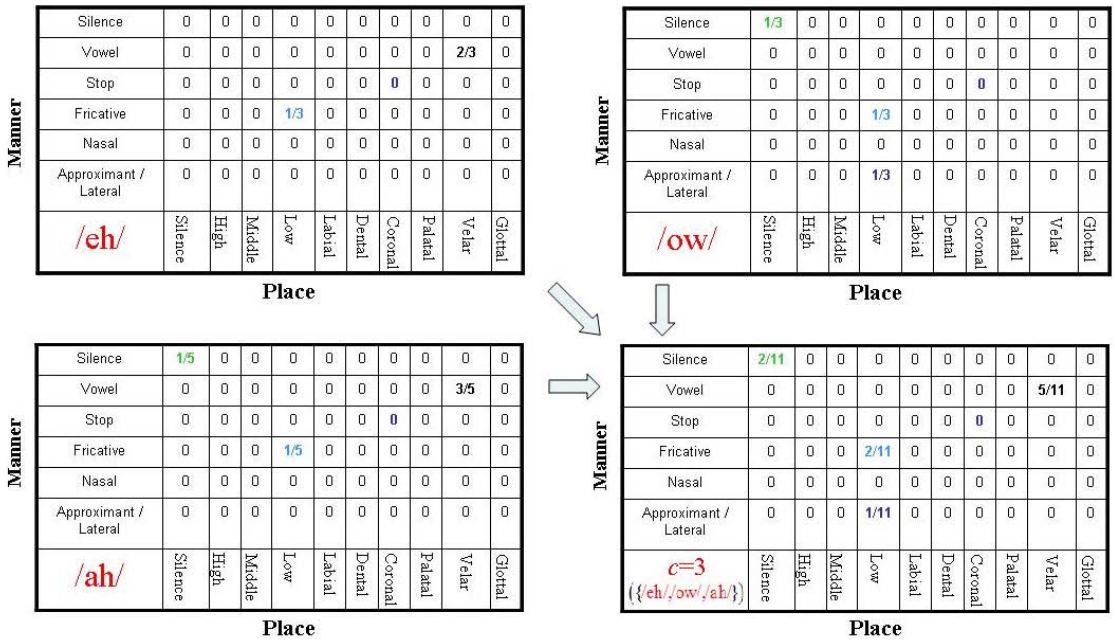


Table 3.5: The phoneme-dependent and phonetic-class dependent AFCPMs correspond to phoneme /eh/, /ah/, /ow/ and the third phonetic class ($c = 3$). The models were obtained by using data shown in Table 3.4 and Eqs. 3.2 and 3.7.

where $c \in \{1, \ldots, G\}$ and r_w is a relevant factor. These coefficients will be used for weighting the phonetic-class dependent speaker models (see Section 3.3.3 below).

3.3.3 Phonetic-Class Dependent Speaker Models

A phonetic-class speaker model can be obtained from speaker-dependent data as follows (see Figure 3.7):

$$\begin{aligned} & P_s^{\mathrm{CD}}(m, p|c) \\ &= P_s^{\mathrm{CD}}(L^{\mathrm{M}} = m, L^{\mathrm{P}} = p|\mathrm{PhoneClass} = c, \mathrm{Speaker} = s) \\ &= \frac{\#((m, p, c) \text{ in the utterances of speaker } s)}{\#((*, *, c) \text{ in the utterances of speaker } s)}. \end{aligned} \tag{3.9}$$

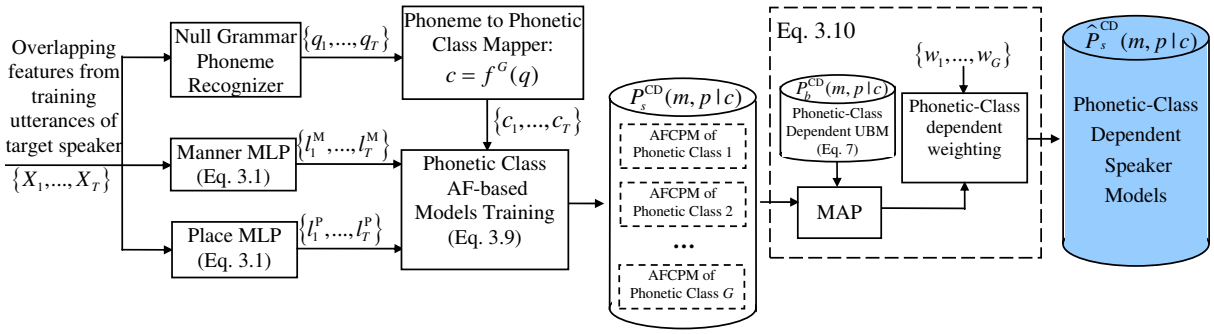


Figure 3.7: The procedure of training the phonetic class AF-based speaker models.

Similar to the phoneme-dependent case, MAP adaptation is applied to obtain the final speaker model:[3]

$$\widehat{P}_s^{\mathrm{CD}}(m,p \mid c) \;=\; \beta_c w_c P_s^{\mathrm{CD}}(m,p \mid c) + (1-\beta_c) w_c P_b^{\mathrm{CD}}(m,p \mid c) \tag{3.10}$$

where, $\beta_c \in [0,1]$ is a phonetic class-dependent adaptation coefficient controlling the contribution of the speaker data and the background models (Eq. 3.7) on the MAP-adapted model. It is obtained by

$$\beta_c = \frac{\#((*,*,c) \text{ in the utterances of speaker } s)}{\#((*,*,c) \text{ in the utterances of speaker } s) + r_\beta} \tag{3.11}$$

where r_β is a fixed relevance factor common to all phonetic classes and speakers. Its purpose is to control the dependence of the adapted model on speaker's data.

Because for each speaker, the accuracy of his/her phonetic-class models depends on the amount of training data for estimating the mapping functions, it is intuitive to weight the density functions by the weighting coefficients w_c in Eq. 3.10. Alternatively we may also train an MLP to optimally weight the phonetic-class, as in [38].

Figure 3.8 shows the background model for phonetic class $c = 3$ of which phonemes

[3] Although strictly speaking $\widehat{P}_s^{\mathrm{CD}}(m,p \mid c)$ is not probability because of the weighting factor w_c, we use the symbol $\widehat{P}$ here for readability and consistency.

/ah/ and /ow/ in Figure 3.3 are members. Also shown are the phonetic-class speaker models of speakers 1018 and 3823 in NIST00. We can observe from Figures 3.8(b) and 3.8(c) that the two speaker models become more distinct (therefore more discriminative) when compared with the phoneme-dependent speaker models in Figure 3.3. The Euclidean distance d between the phonetic-class speaker models (Figures 3.8(b) and 3.8(c)) is also larger than that of the phoneme-dependent models (Figures 3.3 (c)–(f)): 11.08 vs. 8.34 and 7.36. Moreover, the distances between the speaker models and the background models are also larger in the phonetic-class case, primarily because of more data are available for training the phonetic-class speaker models. All of these results suggest that phonetic-class dependent speaker models are more discriminative.

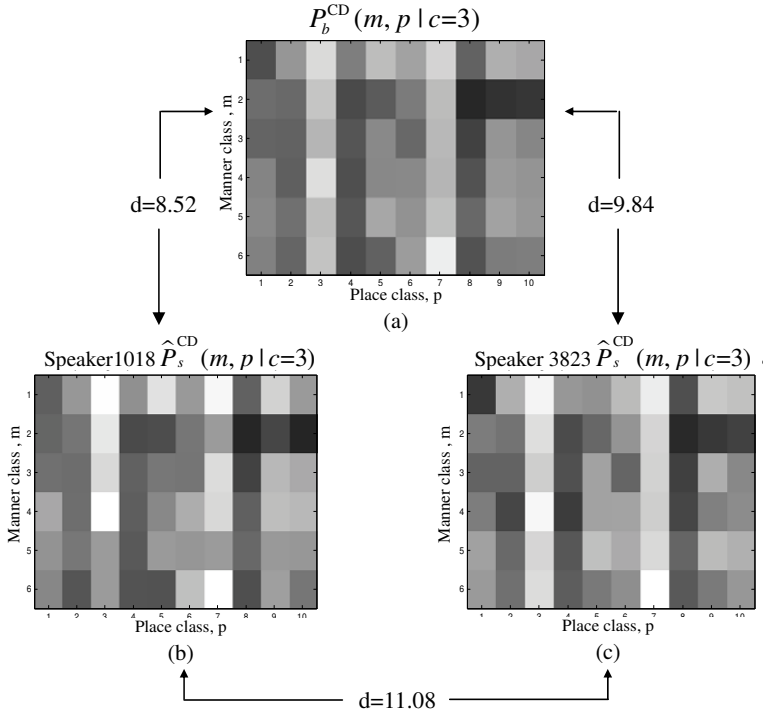


Figure 3.8: Phonetic-class dependent models in which the phonemes /ah/ and /ow/ are members of the phonetic class ($c = 3$ in Table 3.3). The speaker models were obtained from the training utterances of speakers 1018 and 3823 in NIST00, using the mapping function $f^G_{\mathrm{P+VQ}}(q)$. d represents the Euclidean distance between the models pointed to by arrows. The 60 discrete probabilities corresponding to the combinations of the 6 manner and 10 places classes are nonlinearly quantized to 256 gray levels using log scale, where white represents 0 and black represents 1. The 6 manner and 10 places classes in ascending order of the axis labels are : {Silence, Vowel, Stop, Fricative, Nasal, Approximant-Lateral} and {Silence, High, Middle, Low, Labial, Dental, Coronal, Palatal, Velar, Glottal}.

3.3.4 Scoring Method

Following the scoring method in [35], we define the verification score of a test utterance $X = \{X_1, \ldots, X_t, \ldots, X_T\}$ as:

$$S_{\text{CD-AFCPM}}(X) = \frac{1}{T}\sum_{t=1}^{T}\left[\log \widehat{p}_s^{\,\mathrm{CD}}(X_t) - \log p_b^{\mathrm{CD}}(X_t)\right] \tag{3.12}$$

where the speaker models (Eq. 3.10) and background models (Eq. 3.7) are used to compute the scores

$$\begin{aligned}\widehat{p}_s^{\,\mathrm{CD}}(X_t) &= \widehat{P}_s^{\mathrm{CD}}(l_t^{\mathrm{M}}, l_t^{\mathrm{P}}|c_t) \\ &= \widehat{P}_s^{\mathrm{CD}}(L^{\mathrm{M}} = l_t^{\mathrm{M}}, L^{\mathrm{P}} = l_t^{\mathrm{P}}|\mathrm{PhoneClass} = c_t, \mathrm{Speaker} = s)\end{aligned} \tag{3.13}$$

and

$$\begin{aligned}p_b^{\mathrm{CD}}(X_t) &= P_b^{\mathrm{CD}}(l_t^{\mathrm{M}}, l_t^{\mathrm{P}}|c_t) \\ &= P_b^{\mathrm{CD}}(L^{\mathrm{M}} = l_t^{\mathrm{M}}, L^{\mathrm{P}} = l_t^{\mathrm{P}}|\mathrm{PhoneClass} = c_t, \mathrm{Background}),\end{aligned} \tag{3.14}$$

where $c_t = f^G(q_t)$ is the phonetic class of frame t.

For the acoustic GMM system, we applied feature transformation [39] to reduce the effect of channel distortion. Then, acoustic scores S_{GMM} were computed based on GMM-UBM framework [6]:

$$S_{\mathrm{GMM}}(X) \;=\; \frac{1}{T}\sum_{t=1}^{T}\left[\log p(X_t|\Lambda_s) - \log p(X_t|\Lambda_b)\right] \tag{3.15}$$

where Λ_s and Λ_b are the acoustic GMM of speaker s and the acoustic UBM, respectively.

3.4 Experiments

3.4.1 Speech Corpora and Features

NIST99 [40], NIST00 [41], SPIDRE [42], and HTIMIT [43] were used in the experiments. The NIST99 was used for creating the background models and mapping functions, and NIST00 was used for creating speaker models and for performance evaluation. HTIMIT and SPIDRE were used for training the AF-MLPs and the null-grammar phone recognizer, respectively. The purposes of the databases used in this work are summarized in Table 3.6.

Database	Purpose
SPIDRE	To train the null-grammar phone recognizer
HTIMIT	To train the AF-MLPs
NIST99	To create the background models and mapping functions
NIST00	To create speaker models and evaluate their performance

Table 3.6: The purposes of the databases used in this study.

NIST00 contains landline telephone speech extracted from the SwitchBoard-II, Phase 1 and Phase 4 Corpora. The evaluation set comprises 457 male and 546 female target speakers. For each speaker, approximately 2 minutes of speech is available for enrollment. There are 3,026 female and 3,026 male verification utterances. Each verification utterance has length not exceeding 60 seconds and was evaluated against 11 hypothesized speakers of the same sex as the speaker of the verification utterance. This amounts to 6,096 speaker trials and 60,476 impostor attempts.

The acoustic features for training the HMMs and speaker models are slightly different. For the HMMs, acoustic vectors of 39 dimensions—each comprising of 12 Mel-frequency cepstral coefficients (MFCCs) [44], the normalized energy, and their first- and second-order derivatives—were used. For the MFCC-based and AFCPM-based speaker models, 19 mean-normalized MFCCs and their first-order derivatives were computed every 10ms using a Hamming window of 25ms. The MFCCs and delta MFCCs were concatenated to form 38-dimensional feature vectors. Cepstral mean subtraction (CMS), fast blind stochastic features transformation (fBSFT) [39], [45] and short-time Gaussianization (STG) were applied to the MFCCs to remove channel effects.

3.4.2 Training Procedures

3,794 utterances selected from HTIMIT were used to train the manner and place MLPs, and utterances from SPIDRE were used to train a null-grammar phoneme recognizer with 46 context-independent phoneme models (3-state 16-mixture HMMs).

The training part of NIST99 was used to create gender-dependent acoustic (MFCC-based) background models with 1024 mixtures. The same set of data was also used to build phoneme-dependent and phonetic-class dependent AF-based UBMs, which are subsequently used for obtaining the gender-dependent mapping functions using the three methods mentioned in Section 3.3.1. Then, for each target speaker in NIST00, his/her speaker models were created using Eq. 3.10 and the 2-minute enrollment speech based on the mapping functions and the phonetic-class dependent UBMs.

3.4.3 Fusion of MFCC- and AFCPM-Based Systems

Research has shown that features and classifiers of different types may complement each other, and thus improvement in classification performance can be obtained by fusing them [31, 46]. There are several types of fusion for speaker verification, e.g, feature-level fusion and decision-level fusion. Decision-level fusion includes abstract fusion and score fusion. This dissertation focuses on the score-level fusion.

The phonetic-class AFCPMs and the acoustic GMMs characterize speakers at two different levels. The former represents the pronunciation behaviors of individual speakers, whereas the latter focuses on their vocal tract characteristics. Therefore, fusing their scores is expected to improve speaker verification performance. In this work, we have tried three score-level fusion approaches: linear fusion, polynomial fusion (using 2-order polynomial function), and Decision-Based Neural Networks (DBNN) [47] fusion.

Viewing the fusion from another perspective, finding a good fusion function amounts to finding the best decision boundary to separate the genuine speaker scores and

impostor scores in the GMM-UBM versus CD-AFCPM score space, as illustrated in Figure 3.9. The figure shows that the decision boundaries near the high density region of the score space have a similar shape, suggesting that the three fusion approaches should achieve more or less the same performance. Empirically, our experiments on fusion suggest that the three fusion methods produce almost identical EER. Therefore, this dissertation only focuses on linear fusion.

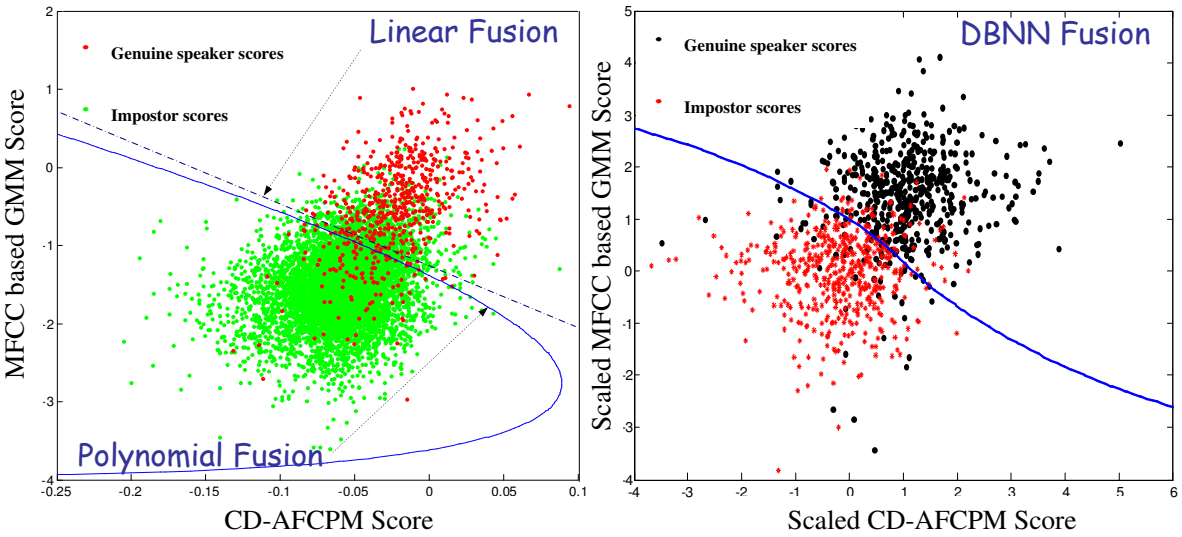


Figure 3.9: Linear, polynomial, and DBNN fusion. Distribution of the score vectors from an MFCC-based GMM-UBM system and a CD-AFCPM system for the first 10% of genuine and impostor trials in NIST00.

In linear score fusion, the utterance scores $S_{\text{CD-AFCPM}}$ and S_{GMM} obtained from phonetic-class dependent AFCPM system and an acoustic GMM-UBM system are linearly combined to obtain a fused score:

$$S_{\text{F}}(X) \;=\; \alpha_u S_{\text{CD-AFCPM}}(X) + (1-\alpha_u) S_{\text{GMM}}(X) \tag{3.16}$$

where $\alpha_u \in [0,1]$ is a fusion weight determined by minimizing the detection cost function (DCF) on training data.

3.5 Results and Discussion

3.5.1 Comparing Different Mapping Functions

Table 3.7 shows the EERs obtained by phoneme-dependent AFCPM (PD-AFCPM) and phonetic-class dependent AFCPM (CD-AFCPM) using the three phoneme-to-phonetic class mapping functions. It shows that the mapping function $f^G_{\mathrm{P+VQ}}(q)$ achieves the lowest error rates in CD-AFCPM. This result suggests that phone properties and Euclidean distance between AF models (VQ) play a complementary role. We conjectures that the phone properties constrain the possible partitioning of phonemes and VQ provides a fine division within the phoneme groups where phone properties alone cannot entirely represent the articulatory properties of speech. In particular, for some large phoneme groups (e.g., vowels), it may be better to partition the groups into subgroups based on the distribution of the AF models than to divide the groups based purely on their phone properties. Completely relying on the distribution of AF models, however, is inappropriate because some constraints are essential for forming the large phoneme groups.

For each mapping function, we also compare class-weighted scoring and equally-weighted scoring (i.e. $w_c = 1$ in Eq. 3.10 for all c). Table 3.7 shows that using class-weighted scores is consistently better than using equally-weighted scores.

3.5.2 Comparing CD-AFCPM and PD-AFCPM

Table 3.7 also shows that phonetic-class AFCPM, regardless of the type of mapping functions, is superior to phoneme-dependent AFCPM. This confirms our earlier argument that when the amount of enrollment data is limited, we had better to enrich the amount of training data per model by grouping similar phonemes together. We advocate phonetic-class dependent AFCPMs (especially the one that uses mapping function $c = f^G_{\mathrm{P+VQ}}(q)$) for two reasons. First, unlike phoneme-dependent AFCPM where training data are divided into 46 classes, data are divided into a maximum of

13 classes only in phonetic-class dependent AFCPM. As a result, a lot more data are available for training each phonetic-class dependent AFCPM, which leads to more reliable speaker models under limited enrollment data. Second, because of the small number of classes, phonetic-class dependent AFCPM is less sensitive to the accuracy of the phoneme recognizer. In phoneme-dependent AFCPM, acoustically confusable phonemes may cause the phoneme recognizer to make mistakes, leading to erroneous scores. However, some of the confusable phonemes may be mapped to the same phonetic class in phonetic-class dependent AFCPM, which effectively alleviate the effect caused by phoneme recognition errors. There seems to be a tradeoff between the number of models per speaker and the representation ability of the models. In particular, a large number of models (e.g., 46 in PD-AFCPM) could lead to inferior performance, as evident in Table 3.7.

The p-values [48] between the EERs obtained by PD-AFCPM and that by all of the CD-AFCPM are less than 0.00001, suggest that the differences in EERs are statistically significant.

3.5.3 Results on Fusing High- and Low-level Features

Let us take a closer look at the fusion between high-level articulatory features and low-level acoustic features. Table 3.8 shows that the UBM-GMM system that uses acoustic features as inputs achieves a significantly lower error rate as compared to the system that uses high-level features. The inferiority of high-level features is primarily due to the short verification utterances (15–45 seconds). However, fusing the scores obtained from these systems can lower the error rates further. The table also shows that fusion of phonetic-class AFCPM and GMM outperforms the fusion of phoneme-dependent AFCPM and GMM. The lowest error rate is achieved by fusing CD-AFCPM and GMM where the low-level features have been transformed by short-time Gaussianization and blind stochastic feature transformation. The p-values between the EER obtained by PD-AFCPM and that by CD-AFCPM are less than

0.00001. This suggests that fusion of low- and high-level features can bring significant performance gain, although the gain diminishes progressively when the low-level features become more robust. Making the low-level features robust, however, does not come without a price. It has been shown recently that using STG and fBSFT as feature preprocessors requires 52 seconds to process a 53-second utterance on a Pentium IV 3.2GHz CPU, whereas processing the same utterance by CMS alone takes only 0.02 seconds [45].

The DET plots corresponding to Table 3.7 and Table 3.8 are shown in Figure 4.6. Evidently, the fusion of phonetic-class AFCPM and GMM achieves the best performance across a wide range of decision threshold. It is obvious that the high-level information captured by the phonetic-class dependent AFCPMs complements the short-term spectral information very well.

	Phoneme Categorization Scheme	No. of Classes G	EER(%)			
			female		male	
			Equally weighted	Class weighted	Equally weighted	Class weighted
CD-AFCPM	VQ $c = f_{\text{VQ}}^{G}(q)$	8	26.72	26.42	23.85	23.74
		10	25.22	24.93	23.70	23.65
		12	25.64	25.36	23.73	23.71
	Phone Properties $c = f_{\text{P}}^{G}(q)$	8	25.04	24.85	24.32	24.11
		11	24.13	23.92	23.31	23.24
		13	24.48	24.25	23.09	23.10
	Phone Properties+VQ $c = f_{\text{P+VQ}}^{G}(q)$	12	23.63	**23.46**	22.89	**22.83**
			Class Weighted Mix gender: **23.76**			
PD-AFCPM			**26.35**		**24.66**	
			Class Weighted Mix gender: **25.91**			

Table 3.7: EERs obtained by phoneme-dependent AFCPM (PD-AFCPM) and phonetic-class dependent AFCPM (CD-AFCPM) using three different phoneme-to-phonetic class mapping methods. "Equally weighted" means that $w_c = 1$ in Eq. 3.10 for all c. The p-values between the PD-AFCPM and all of the CD-AFCPM are less than 0.00001.

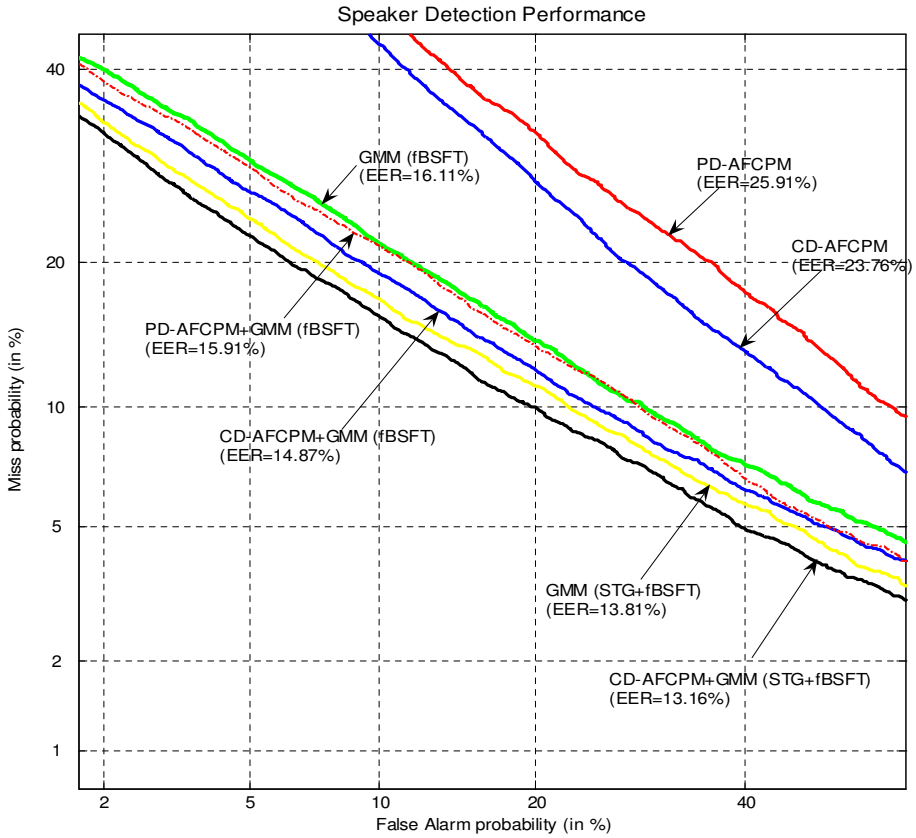


Figure 3.10: DET performance of phonetic-class dependent AFCPM (CD-AFCPM), phoneme-dependent AFCPM (PD-AFCPM), GMM (fBSFT and STG were applied) with mix gender, and their fusions.

3.6 Concluding Remarks

Phoneme-based AFCPM represents the pronunciation characteristics of speakers by building one discrete density function for each phoneme, which requires a large amount of training data to achieve high verification accuracy. Based on the observation that the AFCPMs of some phonemes are very similar, this chapter proposes a speaker verification system that uses phonetic class-based articulatory pronunciation models. Specifically, speaker models are represented by conditional probabilities of articulations given phonetic classes instead of phonemes. Three mapping functions that specify the relationship between phonemes and phonetic classes are proposed. Results

Fusion Results (EER in %)		Acoustic Model	
		GMM(fBSFT)	GMM(STG+fBSFT)
Pronunciation Model	None	16.11	13.81
	PD-AFCPM	15.91	13.71
	CD-AFCPM	14.87	13.16

Table 3.8: EERs obtained by acoustic GMM, phoneme-dependent AFCPM (PD-AFCPM) + GMM, and phonetic-class dependent AFCPM (CD-AFCPM) + GMM. The EERs corresponding to CD-AFCPM are based on class-weighted mixed-gender scenario (see Table 3.7). Note that the fusion of phonetic-class AFCPM and GMM is based on the phonetic-class AFCPM that uses the mapping function $f^{G}_{\mathrm{P+VQ}}$. The p-values between PD-AFCPM+GMM and CD-AFCPM+GMM are less than 0.00001.

show that among the three mapping functions, the one that combines the classical phoneme tree and Euclidean distance between AFCPMs achieves the best performance. Results also show that phonetic-classes AFCPM achieves a significantly lower error rate as compared to conventional AFCPM.

Chapter 4

PROBABILISTIC-WEIGHTED PHONETIC-CLASS AFCPM

Although Chapter 3 has shown that CD-AFCPM is a promising approach to high-level speaker verification, the method still has plenty of room for improvement. This chapter proposes two approaches to further improve the performance of CD-AFCPM. The results show that small but statistically significant improvement can be obtained by applying the proposed approaches.

4.1 Introduction and Motivation

To improve the accuracy of articulatory feature-based models, Chapter 3 proposes to group similar phonemes into phonetic classes by using a mapping function and to represent the background and speaker models as phonetic-class dependent density functions. The mapping function uses hard-decision VQ (see Section 3.3.1). In other words, each AFCPM vector or phoneme is categorized into one of the phonetic classes regardless of its proximity to other classes. This hard-decision based mapping function is simple and fast, but it ignores the possibility that the AFCPM vector may also belong to other classes. Performance may be improved by incorporating the class membership of each phoneme-dependent AFCPM vector in the mapping function. This chapter proposes a CD-AFCPM in which speaker models are created from a probabilistic phoneme-to-phonetic class mapping function. The resulting model is referred to as the probabilistic-weighted CD-AFCPM, or simply PW-CD-AFCPM.

A new scoring method that uses an SVM to combine the scores obtained from

different phonetic-class dependent models is also proposed.

4.2 Probabilistic Weighted Phonetic-Class Dependent AFCPM (PW-CD-AFCPM)

4.2.1 Mapping Function

Because the results in Chapter 3 show that the mapping function that uses VQ and phone properties (i.e. $f^G_{\text{P+VQ}}$) achieves the best performance, this chapter focuses on this mapping function only.

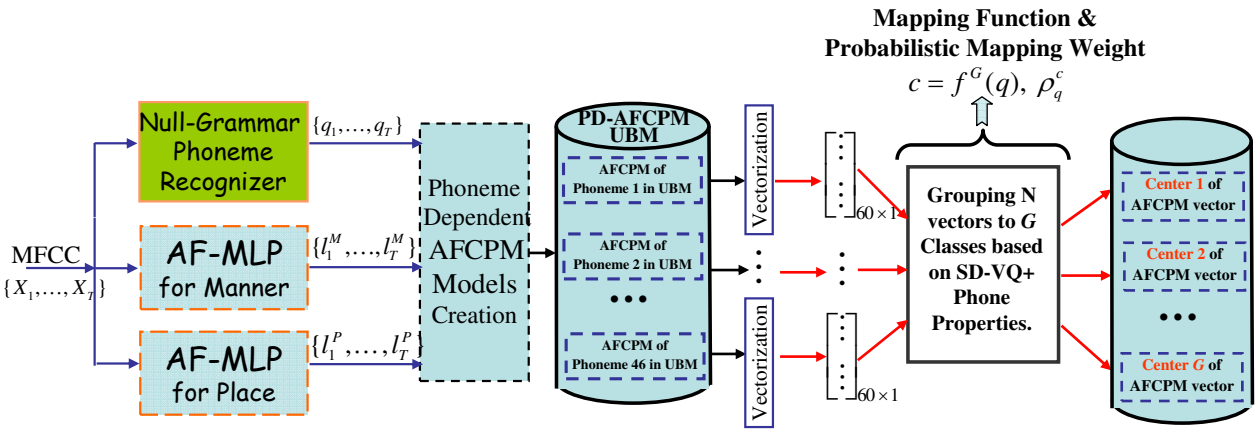


Figure 4.1: Training the mapping function and mapping weights. SD-VQ stands for soft-decision VQ.

4.2.2 Probabilistic Mapping Weights

Without loss of generality, let's define the i-th phoneme group as $\mathcal{C}_i = \{\mathcal{C}_i^1, \ldots, \mathcal{C}_i^{N_i}\}$, where $\mathcal{C}_i^j$ is the j-th phonetic class created by applying VQ to the AFCPM vectors $\mathbf{a}_q$ in $\mathcal{C}_i$. Therefore, we have $\sum_{i=1}^{8} N_i = G$, where G is the number of phonetic classes. Each phonetic class c has a centre vector $\mathbf{m}_c$, where $c = f^G_{\text{P+VQ}}(q)$ and $c = \{1, \ldots, G\}$. Let us denote $\rho^c_{q_t} \equiv P(c|q_t)$ as the probability of phoneme q_t belonging to phonetic class c, which can be approximated by $P(c|\mathbf{a}_{q_t})$. Let's also assume that the distribution

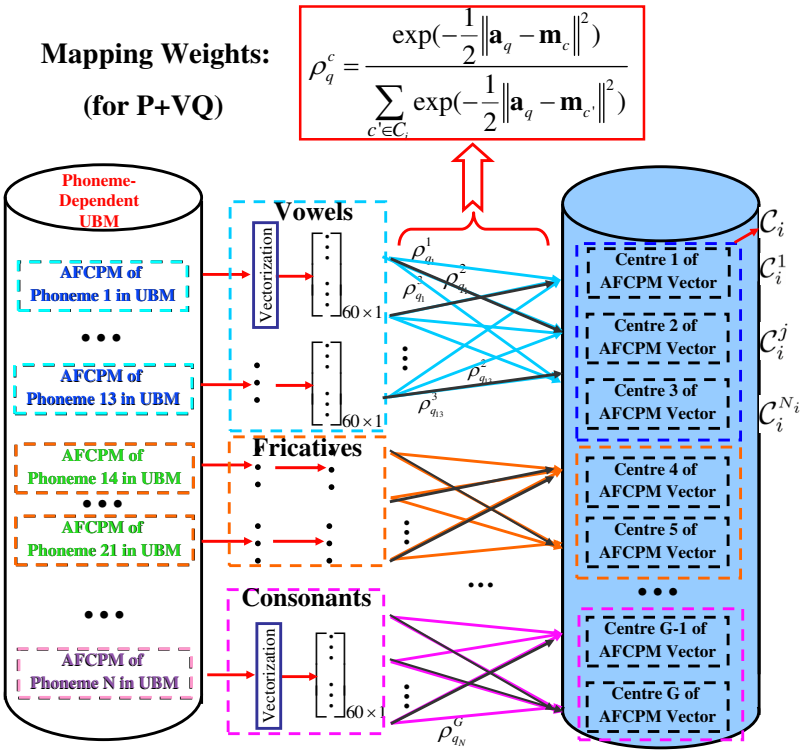


Figure 4.2: The procedure of training the probabilistic mapping weights and mapping function.

$P(\mathbf{a}_{q_t}|c)$ is a Gaussian function with mean $\mathbf{m}_c$ and covariance $\Sigma_c = \mathbf{I}$ and that the prior probabilities $P(c)$ are equal for all classes. Therefore, the mapping weights $\rho_{q_t}^c$ for phoneme q_t and phonetic class $c \in \mathcal{C}_i$ can be computed as follows:

$$\begin{aligned}
\rho_{q_t}^c \equiv P(c|q_t) \approx P(c|\mathbf{a}_{q_t}) &= \frac{P(\mathbf{a}_{q_t}|c)P(c)}{\sum_{c' \in \mathcal{C}_i} P(\mathbf{a}_{q_t}|c')P(c')} \\
&\approx \frac{\frac{1}{(2\pi)^{d/2}|\Sigma_c|} \exp\left\{-\frac{1}{2}(\mathbf{a}_{q_t} - \mathbf{m}_c)^T {\Sigma_c}^{-1}(\mathbf{a}_{q_t} - \mathbf{m}_c)\right\}}{\sum_{c' \in \mathcal{C}_i} \frac{1}{(2\pi)^{d/2}|\Sigma_{c'}|} \exp\left\{-\frac{1}{2}(\mathbf{a}_{q_t} - \mathbf{m}_{c'})^T {\Sigma_{c'}}^{-1}(\mathbf{a}_{q_t} - \mathbf{m}_{c'})\right\}} \\
&= \frac{\exp\left(-\frac{1}{2}\|\mathbf{a}_{q_t} - \mathbf{m}_c\|^2\right)}{\sum_{c' \in \mathcal{C}_i} \exp\left(-\frac{1}{2}\|\mathbf{a}_{q_t} - \mathbf{m}_{c'}\|^2\right)} .
\end{aligned} \tag{4.1}$$

where $\mathcal{C}_i$ represents the phonetic classes in the i-th group. Note that $P(c|\mathbf{a}_{q_t})$ is a monotonically decreasing function of $\|\mathbf{a}_{q_t} - \mathbf{m}_c\|$, where $\mathbf{m}_c$ is the centroid of phonetic class c.

The procedures of training the mapping function and the probabilistic mapping weights are illustrated in Figure 4.1.

4.2.3 Probabilistic-Weighted Phonetic-Class Dependent UBMs

Given a mapping function, the phonetic-class dependent UBMs of phonetic class c proposed in Chapter 3 can be written as:

$$\begin{aligned} & P_b^{\mathrm{CD}}(m,p|c) \\ &= P_b^{\mathrm{CD}}(L^{\mathrm{M}}=m, L^{\mathrm{P}}=p|\mathrm{PhoneClass}=c, \mathrm{Background}) \\ &= \frac{\#((m,p,c)\text{in the untterances of all background speakers})}{\#((*,*,c)\text{in the untterances of all background speakers})} \\ &= \frac{\sum_{t\in\mathcal{T}_{\mathrm{b}}} 1}{\sum_{t\in\mathcal{T}_{\mathrm{b}}'} 1}\,, \qquad m\in\mathcal{M}, p\in\mathcal{P}, c\in\{1,\ldots,G\} \end{aligned} \tag{4.2}$$

where $\mathcal{T}_{\mathrm{b}} = \{t : l_t^{\mathrm{M}} = m, l_t^{\mathrm{P}} = p, f^G(q_t) = c,\ X_t \in \text{all background speakers}\}$, $\mathcal{T}_{\mathrm{b}}' = \{t : f^G(q_t) = c, X_t \in \text{all background speakers}\}$, L^{M} and L^{P} represent the manner and place labels, respectively, and l_t^{M} and l_t^{M} are the manner and place labels determined by the manner and place MLPs, respectively. Eq. 4.2 suggests that all frames are weighted equally. However, frames that have a higher probability of belonging to phonetic class c should be given a higher weight and vice versa for frames that have a lower probability. Therefore, it is intuitive to weight the contribution of frame t as follows:

$$\begin{aligned} & P_b^{\mathrm{CD}}(m,p|c) \\ &= P_b^{\mathrm{CD}}(L^{\mathrm{M}}=m, L^{\mathrm{P}}=p|\mathrm{PhoneClass}=c, \mathrm{Background}) \\ &= \frac{\sum_{t\in\mathcal{T}_{\mathrm{b}}} \rho_{q_t}^c}{\sum_{t\in\mathcal{T}_{\mathrm{b}}'} \rho_{q_t}^c}\,, \end{aligned} \tag{4.3}$$

where $\rho_{q_t}^c \equiv P(c|q_t)$ is the probability of phoneme q_t belonging to phonetic class c, which can be approximated by Eq. 4.1.

4.2.4 Probabilistic-Weighted Phonetic-Class Dependent Speaker Models

Target speaker models are obtained in two steps. In the first step, we compute:

$$\begin{aligned} & P_s^{\mathrm{CD}}(m,p|c) \\ &= P_s^{\mathrm{CD}}(L^{\mathrm{M}}=m, L^{\mathrm{P}}=p|\mathrm{PhoneClass}=c, \mathrm{Speaker}=s) \\ &= \frac{\sum_{t\in\mathcal{T}_{\mathrm{s}}}\rho_{q_t}^c}{\sum_{t\in\mathcal{T}_{\mathrm{s}}'}\rho_{q_t}^c}, \quad m\in\mathcal{M}, p\in\mathcal{P}, c\in\{1,\ldots,G\} \end{aligned}$$

where $\mathcal{T}_{\mathrm{s}} = \{t : l_t^{\mathrm{M}} = m, l_t^{\mathrm{P}} = p, f^G(q_t) = c,\ X_t \in \text{speaker } s\}$ and $\mathcal{T}_{\mathrm{s}}' = \{t : f^G(q_t) = c,\ X_t \in \text{speaker } s\}$. Then in the second step, MAP adaptation is applied to obtain the model of target speaker s:

$$\widehat{P}_s^{\mathrm{CD}}(m,p|c) = \beta_c P_s^{\mathrm{CD}}(m,p|c) + (1-\beta_c)P_b^{\mathrm{CD}}(m,p|c) \tag{4.4}$$

where, $\beta_c \in [0,1]$ is a phonetic class-dependent adaptation coefficient controlling the contribution of the speaker data and the background models (Eq. 4.3) on the MAP-adapted model. It is obtained by

$$\beta_c = \frac{\#((*,*,c) \text{ in the utterances of speaker } s)}{\#((*,*,c) \text{ in the utterances of speaker } s) + r_\beta} \tag{4.5}$$

where r_β is a fixed relevance factor common to all phonetic classes and speakers.

The procedure of training a probabilistic-weighted phonetic-class dependent AFCPM (PW-CD-AFCPM) is illustrated in Figure 4.3.

Figure 4.4 shows the background model for phonetic class $c = 3$ of which phonemes /ah/ and /ow/ in Figure 3.3 are members. Also shown are the phonetic-class speaker models of speakers 1018 and 3823 in NIST00. Figures 4.4(b) and 4.4(c) show that the

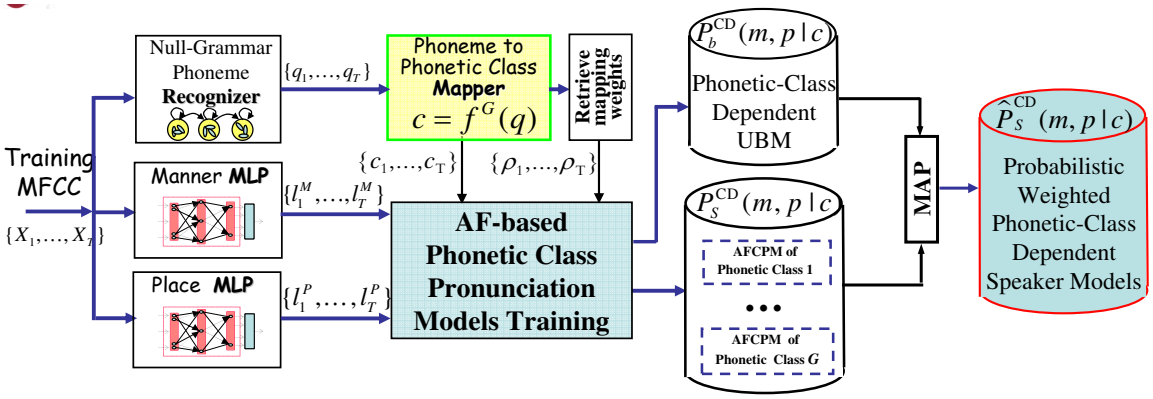


Figure 4.3: The procedure of training a probabilistic-weighted CD-AFCPM.

two probabilistic phonetic-class speaker models are more distinctive (therefore more discriminative) than the phoneme-dependent speaker models shown in Figure 3.3. The Euclidean distance d between the probabilistic phonetic-class speaker models (Figures 4.4(b) and 4.4(c)) is also larger than that of the phoneme-dependent models (Figures 3.3 (c)–(f)): 11.35 vs. 8.34 and 7.36. Moreover, the distances between the speaker models and the background models are also larger in the probabilistic phonetic-class case, primarily because of more data are available for training the phonetic-class speaker models. Even comparing with the CD-AFCPM case (Figure. 3.8), the PW-CD-AFCPMs still have larger distances between the speaker models (11.35 vs. 11.08). All of these results suggest that probabilistic-weighted phonetic-class dependent speaker models are more discriminative.

4.2.5 SVM Scoring

Traditionally, the speaker score is computed by averaging the likelihood ratios in a frame-by-frame basis:

$$S_{\text{CD-AFCPM}}(X) = \frac{1}{T}\sum_{t=1}^{T}\left[\log \widehat{p}_s^{\,\text{CD}}(X_t) - \log p_b^{\text{CD}}(X_t)\right], \tag{4.6}$$

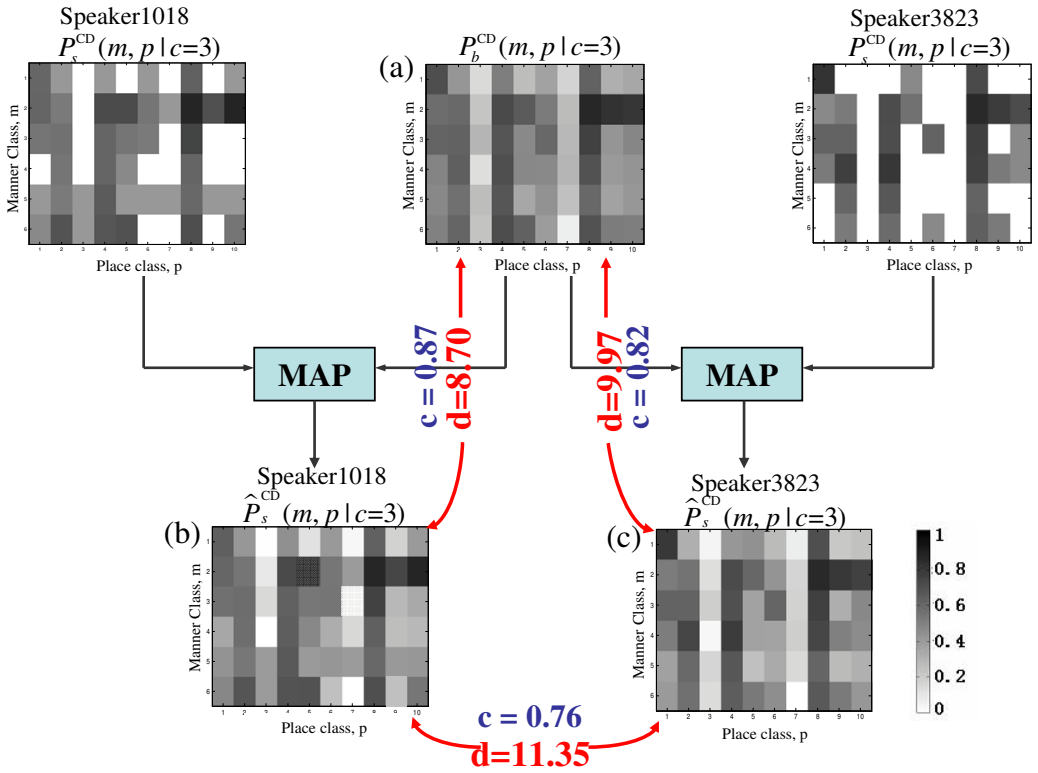


Figure 4.4: Probabilistic-weighted CD-AFCPM in which the phonemes /ah/ and /ow/ are members of the phonetic class. The speaker models were obtained from the training utterances of speakers 1018 and 3823 in NIST00, using the mapping function $f^G_{\text{P+VQ}}(q)$. d represents the Euclidean distance between the models connected by arrows.

where $X = \{X_1, \ldots, X_t, \ldots, X_T\}$ is a test utterance. Note that, we can also express the verification score in Eq. 4.6 as follows:

$$\begin{aligned} S_{\text{CD-AFCPM}}(X) &= \frac{1}{T}\sum_{c=1}^{G}\left(\sum_{t\in\mathcal{T}^c}\left[\log \widehat{p}_s^{\text{CD}}(X_t) - \log p_b^{\text{CD}}(X_t)\right]\right) \\ &= \frac{1}{T}\sum_{c=1}^{G} S^c_{\text{CD-AFCPM}} \end{aligned} \tag{4.7}$$

where $\mathcal{T}^c = \{t : f^G(q_t) = c\}$, and $S^c_{\text{CD-AFCPM}} = \sum_{t\in\mathcal{T}^c}\left[\log \widehat{p}_s^{\text{CD}}(X_t) - \log p_b^{\text{CD}}(X_t)\right]$.

Because ρ_q^c represents the probability of phoneme q belonging to phonetic class c, it makes sense to weight every test frame by ρ_q^c. More specifically, during verification

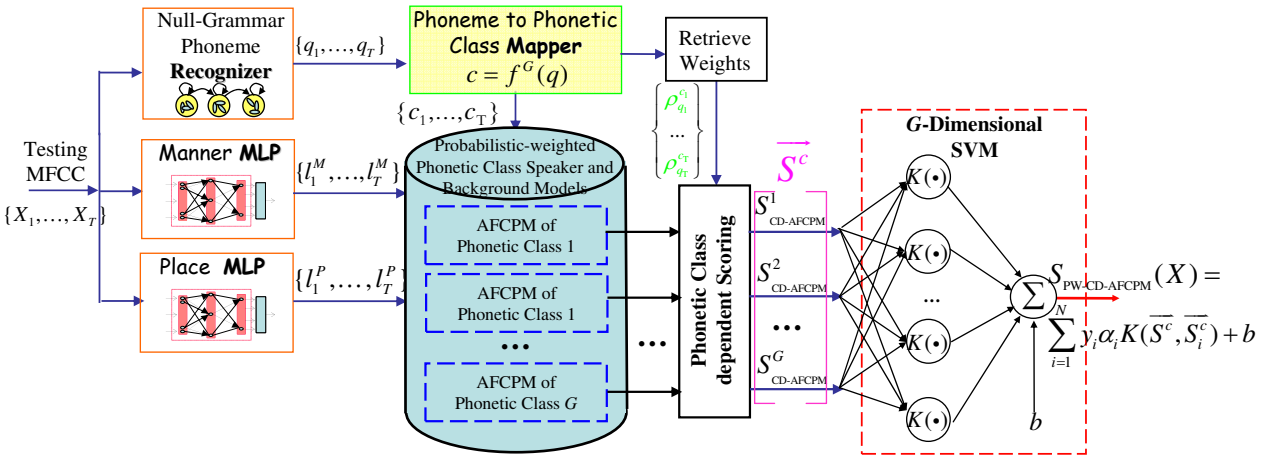


Figure 4.5: The verification phase of probabilistic-weighted CD-AFCPM.

the speaker score $S_{\text{PW-CD-AFCPM}}$ is computed as follows:

$$S_{\text{PW-CD-AFCPM}} = \frac{1}{T}\sum_{t=1}^{T} \rho_{q_t}^{c} \left[\log \widehat{p}_s^{\text{CD}}(X_t) - \log p_b^{\text{CD}}(X_t)\right] \tag{4.8}$$

where $c = f^G(q_t)$. As a result, by applying this probabilistic-weighted strategy, the verification score is computed as:

$$\begin{aligned} S_{\text{PW-CD-AFCPM}}(X) &= \frac{1}{T}\sum_{c=1}^{G}\left(\sum_{t:f^G(q_t)=c} \rho_{q_t}^{c} \left[\log \widehat{p}_s^{\text{CD}}(X_t) - \log p_b^{\text{CD}}(X_t)\right]\right) \\ &= \frac{1}{T}\sum\nolimits_{c=1}^{G} S_{\text{PW-CD-AFCPM}}^{c} \end{aligned}$$

where frame t is weighted by $\rho_{q_t}^{c}$, the probability of phoneme q_t belonging to phonetic class c. The speaker models (Eq. 4.4) and background models (Eq. 4.3) are used to compute the scores

$$\begin{aligned} &\widehat{p}_s^{\text{CD}}(X_t) = \widehat{P}_s^{\text{CD}}(l_t^{\text{M}}, l_t^{\text{P}} | c_t) \\ &= \widehat{P}_s^{\text{CD}}(L^{\text{M}} = l_t^{\text{M}}, L^{\text{P}} = l_t^{\text{P}} | \text{PhoneClass} = c_t, \text{Speaker} = s) \end{aligned}$$

and

$$p_b^{\mathrm{CD}}(X_t) = P_b^{\mathrm{CD}}(l_t^{\mathrm{M}}, l_t^{\mathrm{P}}|c_t)$$
$$= P_b^{\mathrm{CD}}(L^{\mathrm{M}} = l_t^{\mathrm{M}}, L^{\mathrm{P}} = l_t^{\mathrm{P}}|\mathrm{PhoneClass} = c_t, \mathrm{Background}),$$

where $c_t = f^G(q_t)$ is the phonetic class of frame t, and l_t^{M} and l_t^{P} are the AF labels determined by the AF-MLPs.

Eq. 4.9 treats all phonetic classes equally. In general, a summation of scores, as in Eq. 4.9, is likely to give suboptimal solutions. Better results may be obtained by applying an SVM to merge the probabilistic-weighted phonetic-class dependent scores. Specifically, for each utterance, the PW-CD-AFCPM scores ($S^c_{\text{PW-CD-AFCPM}}$) derived from the G phonetic classes form a G-dimensional score vector $\overrightarrow{S^c} = [S^1_{\text{PW-CD-AFCPM}}, S^2_{\text{PW-CD-AFCPM}}, \ldots, S^G_{\text{PW-CD-AFCPM}}]^{\mathrm{T}}$. The vector is then presented to a G-input nonlinear SVM to produce the final verification score:

$$S_{\text{PW-CD-AFCPM}}(X) = \sum\nolimits_{i=1}^{N} y_i \alpha_i K(\overrightarrow{S^c}, \overrightarrow{S_i^c}) + b. \tag{4.9}$$

where $y_i \in \{+1, -1\}$ are class labels of training data, $\overrightarrow{S_i^c} = [S_i^1, S_i^2, \ldots, S_i^G]^{\mathrm{T}}$ $(i = 1, \ldots, N)$ are the training vectors with each dimension representing a probabilistic-weighted phonetic-class score, $K(\cdot)$ is a kernel function, and α_i and b are the SVM parameters that we want to optimize. Figure 4.5 depicts the architecture of a verification system that uses this method.

The nonlinear SVM can be trained and evaluated by using k-fold cross-validation [49]. Specifically, the evaluation data in NIST00 will be divided into k subsets with almost the same number of utterances; for each fold, the speaker and imposter scores from $k-1$ subsets will be used for training a G-input SVM with a second-order polynomial kernel and the remaining subset will be used for evaluation. Therefore, every test utterance will have a chance to be evaluated and the number of scores for evaluation is exactly the same as that in the NIST00 verification protocol.

4.3 Experiments and Results

4.3.1 Procedures

The procedure and data are identical to those in Section 3.4, except that in addition to the conventional scoring method (Eq. 3.12), the results of the SVM scoring method are also reported.

4.3.2 Results and Discussions

Table 4.1 and Table 4.2 show the EERs obtained by CD-AFCPMs and CD-AFCPMs with probabilistic weighting. The results show that using the probabilistic weighting scheme to create CD-AFCPMs can reduce the EER (mixed-gender) from 23.76% to 23.14%, which amounts to 2.61% reduction in error. The p-value between these two EERs is less than 0.0001, suggesting that the difference in EERs is statistically significant.

Modeling Method	EER (%)
CD-AFCPM	Mix gender : **23.76**
PW-CD-AFCPM	Mix gender: **23.14**

Table 4.1: EERs obtained by CD-AFCPM and PW-CD-AFCPM using the mapping function $f^{G}_{\mathrm{P+VQ}}(q)$. The p-values between the two EERs is 0.00005.

Table 4.2 shows the performance of CD-AFCPMs (with and without probabilistic weighting) when they were combined with our best GMM-UBM systems. The results show that small improvement can be achieved by using probabilistic weighting. However, the improvement is not significant. This may be due to the fact that the utterances are too short for AFCPMs, causing the GMM-UBM system to dominate in the verification.

Fusion Results (EER in %)		Acoustic Model	
		GMM(fBSFT)	GMM(STG+fBSFT)
Pronunciation Model	None	16.11	13.81
	PD-AFCPM	15.91	13.71
	CD-AFCPM	14.87	13.16
	PW-CD-AFCPM	14.70	13.09

Table 4.2: EERs obtained by fusing acoustic GMM-UBM systems with PD-AFCPM, CD-AFCPM and PW-CD-AFCPM. The CD-AFCPM and PW-CD-AFCPM use the mapping function $f^{G}_{\mathrm{P+VQ}}(q)$.

Figure 4.6 shows the detection error tradeoff curves corresponding to various low- and high-level systems and their fusion. Apparently, the CD-AFCPMs (with and without probabilistic weighting) and GMM-UBM systems are complementary, and the probabilistic weighting scheme (PW-CD-AFCPM) can help reduce the EER of CD-AFCPM slightly in all operating point. However, when PW-CD-AFCPM is fused with GMM-UBM, only small improvement can be achieved (comparing PW-CD-AFCPM+GMM and CD-AFCPM+GMM).

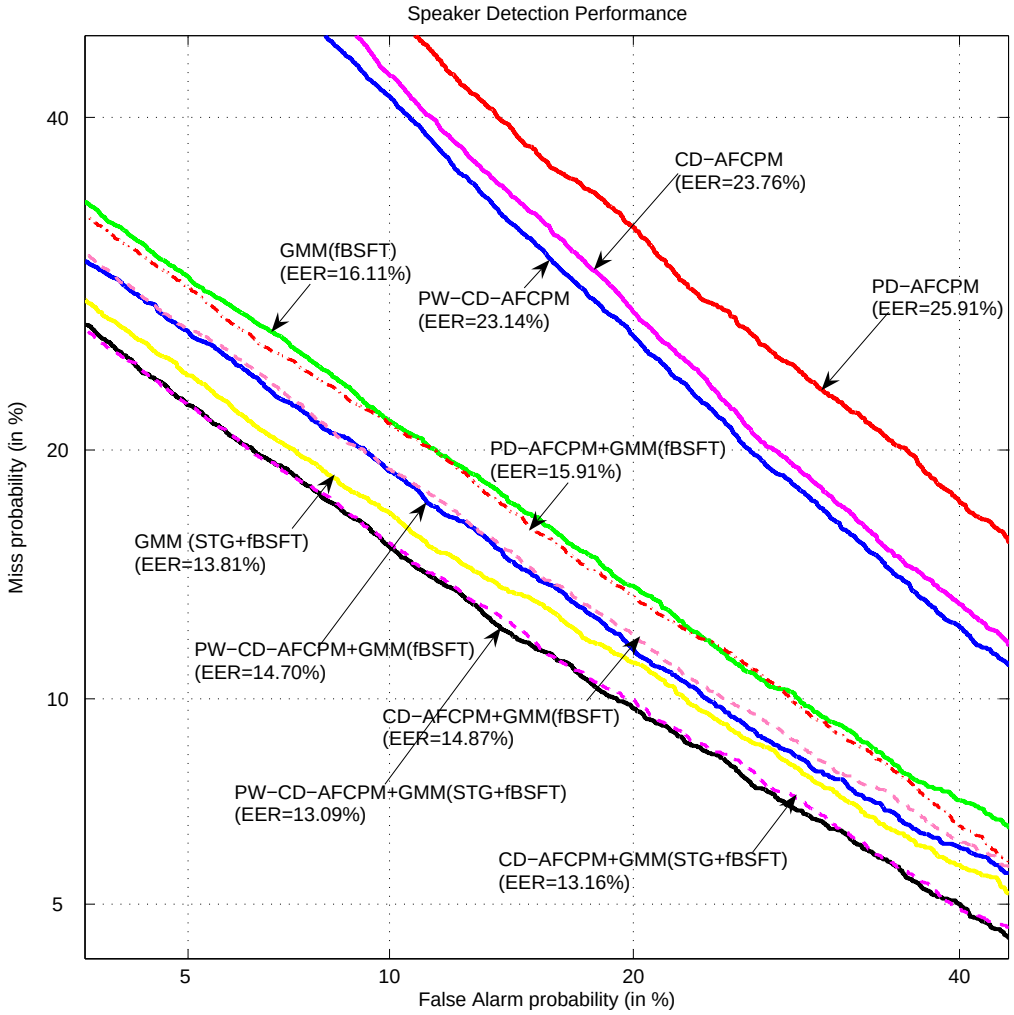


Figure 4.6: DET performance of probabilistic-weighted phonetic-class dependent AFCPM (PW-CD-AFCPM), phoneme-dependent AFCPM (PD-AFCPM), GMM (with fBSFT and STG applied), and their fusions. All curves are based on mixed-gender scores.

Chapter 5

NEW ADAPTATION METHODS FOR SPEAKER-MODEL CREATION IN HIGH-LEVEL SPEAKER VERIFICATION

Research has shown that speaker verification based on high-level speaker features requires long enrollment utterances to be reliable. However, in practical speaker verification, it is common to model speakers based on a limited amount of enrollment data. To minimize the undesirable effect of insufficient enrollment data on system performance, this chapter proposes a new adaptation method for creating speaker models based on high-level features. The proposed method was compared with traditional MAP adaptation under the NIST2000 SRE framework. Experimental results show that the proposed method can solve the data-spareness problem effectively and achieves a better performance when compare with traditional MAP adaptation.

5.1 Introduction and Motivation

Text-independent speaker verification systems typically extract speaker-dependent features from short-term spectra of speech signals to build speaker-dependent Gaussian mixture models (GMMs) [6]. To increase the ability to discriminate between client (target) speakers and impostors, a GMM-based background model is used to represent the characteristics of impostors. The background model can be trained using the speech of non-target background speakers from large speech corpora. Therefore, finding enough speech to train the background model is usually not too difficult. However, obtaining a large number of client utterances is difficult and impractical because most clients are not willing to spend a long time for enrollment.

To address this problem, various adaptation methods, such as maximum a posteri-

ori (MAP) [6], maximum-likelihood linear regression (MLLR) [11], kernel eigen-space MLLR (KEMLLR) [50], and adaptation of phoneme-independent speaker models [51] have been proposed for creating low-level acoustic speaker models from a small amount of client data. It has been shown that KEMLLR outperforms other adaptation methods when the amount of enrollment data is very limited and that when a large amount of enrollment data is available, MAP is a better candidate for creating speaker models [52].

As discussed earlier, using long-term, high-level features to characterize speakers can improve the robustness of speaker verification systems. However, one problem of using high-level features is that it requires a large amount of speech data for creating reliable speaker models. As a result, data-sparseness is a serious problem in high-level speaker verification.

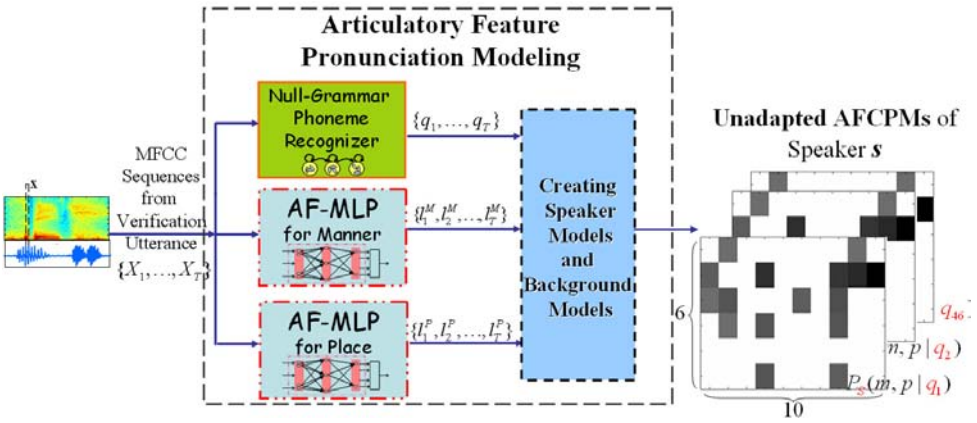


Figure 5.1: Training of unadapted phoneme-dependent AFCPM speaker models and the data-sparseness problem they may encounter.

The simplest way of creating a phoneme-depend AFCPM speaker model is to compute the discrete density function for each phoneme based solely on the speech of the corresponding target speaker, as illustrate in Figure 5.1 (see the detail of phoneme-dependent AFCPM in Section 3.2). However, this naive approach can result in many zero entires in the density functions, primarily because of the data spareness

problem. Although Leung et al. [53] have shown in their articulatory feature-based pronunciation model (AFCPM) that this problem can be tackled by classical MAP adaptation, the client models that they created are essentially a linear weighted sum of enrollment data's distribution and background models. It was found that the modeling capability of the AFCPMs drops rapidly when the amount of enrollment data decreases [54].

To alleviate this problem, this chapter proposes several new adaption and models creation methods as shown in Figure 5.2. Specifically, we propose to adapt not

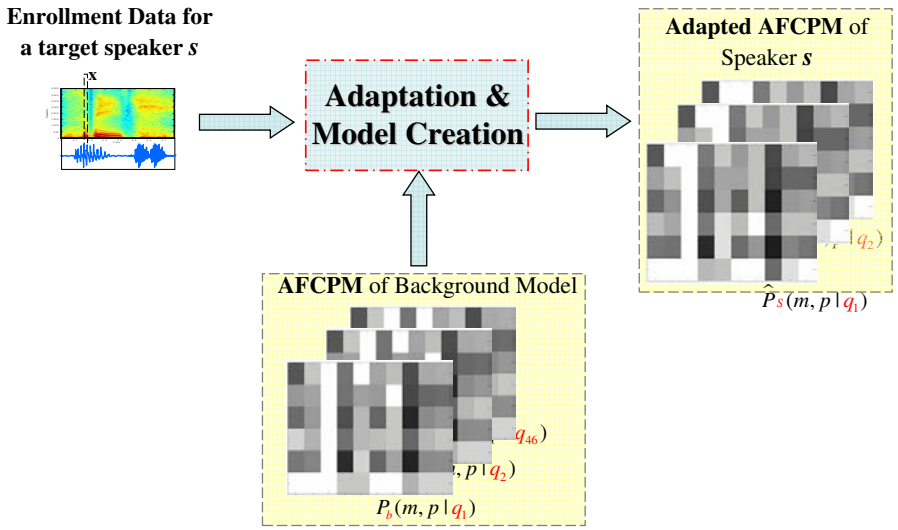


Figure 5.2: The contribution of this chapter: new adaptation methods for speaker-model creation.

only the phoneme-dependent background models but also the phoneme-independent speaker models to create client speaker models. A scaling factor, which is derived from the ratio between the phoneme-dependent background model and the phoneme-independent background model, will also be used to adjust the phoneme-independent speaker models during adaptation.

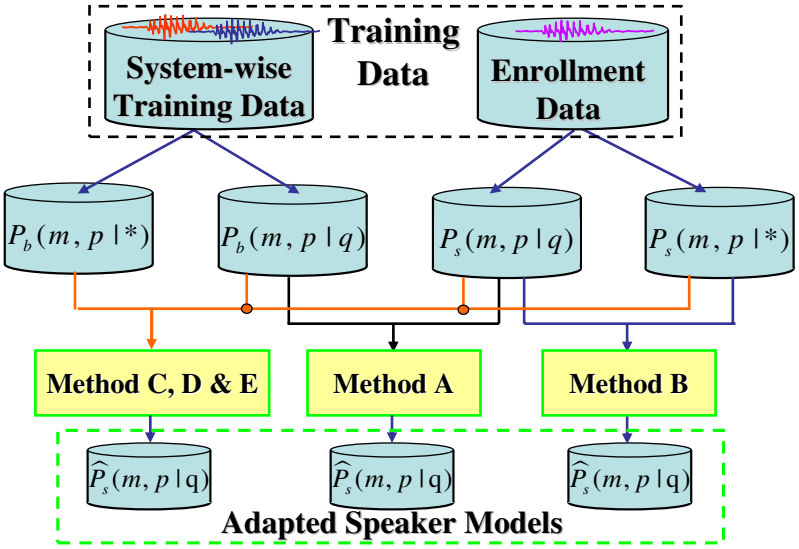


Figure 5.3: Data-set utilization in different adaptation methods. Methods A and B only use part of the available models. Methods C and D fully utilize all of the possible models that can be obtained from training data. '*' means that the corresponding model is phoneme-independent.

5.2 Adaptation Methods for AFCPMs

In this section, we review the classical MAP adaptation and propose four MAP-based adaptation methods that use as much information obtainable from training data as possible (see Fig. 5.3). Five adaptation methods will be discussed.

Method A: Adapted from phoneme-dependent background models (classical MAP used in [53]).

Method B: Adapted from phoneme-dependent speaker models and phoneme-independent speaker models.

Method C: Adapted from phoneme-independent speaker models with a phoneme-dependent scaling factor.

Method D: Adapted from phoneme-dependent background models and phoneme-independent speaker models with a phoneme-dependent scaling factor.

Method E: Adapted from phoneme-independent speaker models and phoneme-dependent background models with a speaker-dependent scaling factor.

5.2.1 Problems of MAP Adaptation for AFCPMs

Method A (classical MAP):

Figure 5.4 illustrates the procedure of applying MAP adaptation (Method A). The adaptation formula is written as:

$$\widehat{P}_s(m,p|q) = \beta_s^q P_s(m,p|q) + (1-\beta_s^q) P_b(m,p|q) \tag{5.1}$$

where $\beta_s^q \in [0,1]$ is a phoneme-dependent adaptation coefficient controlling the con-

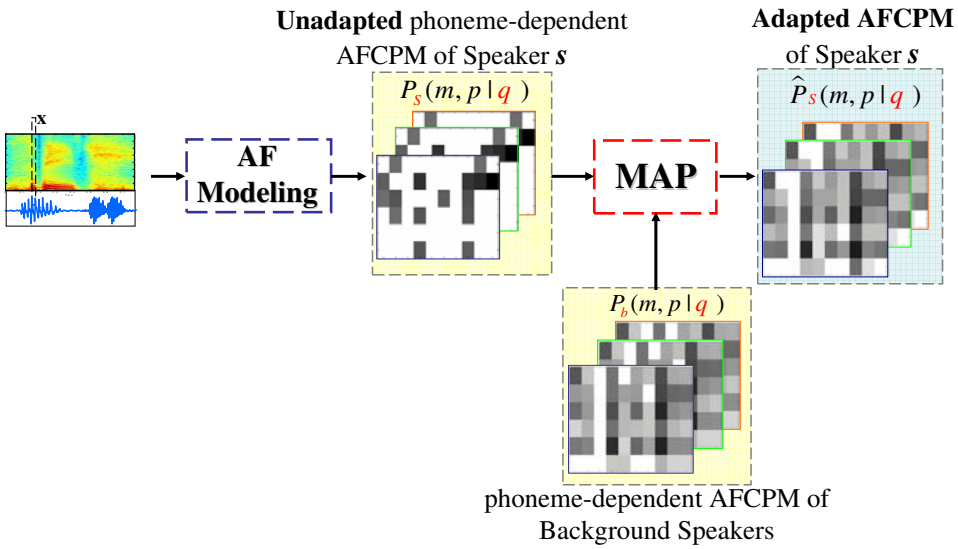


Figure 5.4: The procedure of applying MAP adaptation (Method A) to create phoneme-dependent AFCPM speaker model.

tribution of the enrollment data and the background models (Eq. 3.2) on the MAP-

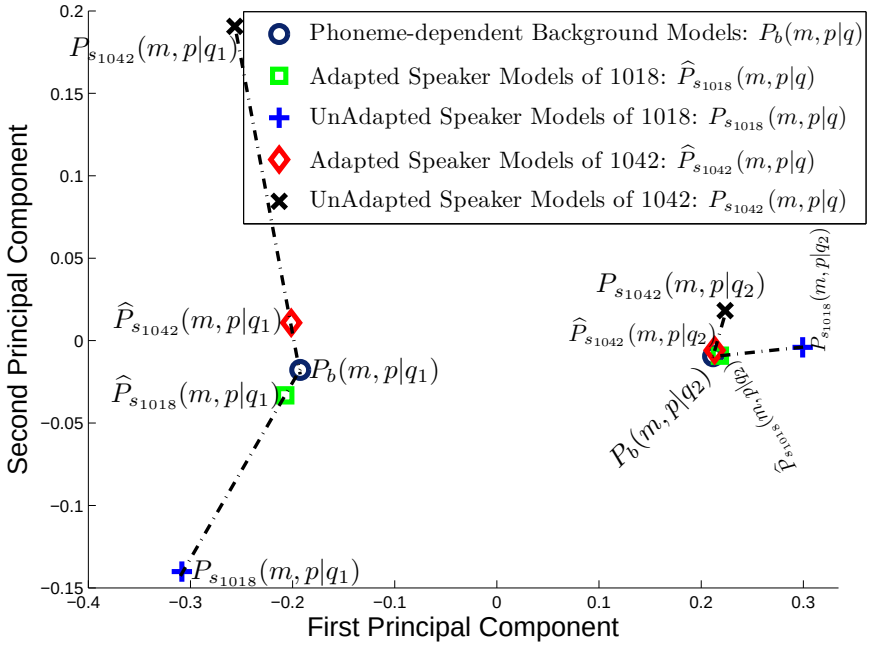


Figure 5.5: *Method A*. Relationship (based on real data) between the background, unadapted, and adapted AFCPMs in classical MAP (q_1=/jh/, q_2=/uw/). The linear combination in Eq. 5.1 suggests that the adapted model will lie along the straight line passing through the unadapted model and the background model.

adapted model (see Section 3.2.1 for details). The relationship between the adapted, unadapted and background models is illustrated in Figure 5.5. The figure shows the relationship between the background, unadapted, and adapted AFCPMs in classical MAP by projecting the AFCPMs onto their first two PCA axes [55]. When enrollment data is sufficient, MAP adaptation can create client models that capture the phoneme-dependent characteristics of speakers. However, when the amount of enrollment data is limited, this speaker-model creation method may have three fundamental problems:

Problem 1: The method will make the client models of the same phoneme too close to the background model of that phoneme, even though the clients may have very different pronunciation characteristics. This will cause the client

models fail to discriminate the true speakers from the imposters.

Problem 2: The method does not fully utilize the information available in the training data.

Problem 3: The method imposes too much constraint on the adaptation.

Problem 1 is exemplified in Fig. 5.6, where the adapted models of two speakers are very similar because they are very close to the background model. Comparison between Figures. 5.6(d) and 5.6(e) reveals that the model of speaker 1018 is very similar to that of speaker 1042. This will make the speaker models fail to discriminate the true speakers from impostors.

For Problem 2, the method only uses two out of four possible models for adaptation. Figure. 5.3 shows the possible models from which the target models can be adapted. Method A uses the phoneme-dependent models only and ignores the fact that the phoneme-independent models ($P_b(m, p|*)$ and $P_s(m, p|*)$) can also be used to create target speaker models.

For Problem 3, the method uses all of the background speakers' data to train phoneme-dependent background models from which phoneme-dependent target speaker models are created by MAP adaptation. Creating a phoneme-dependent speaker model from the corresponding phoneme-dependent background model means that the resulting speaker model is constrained by the articulatory properties of a single phoneme. In other words, the method does not allow cross-phoneme adaptation. Note that the classical MAP adaptation for acoustic GMMs does not have such a hard constraint. Instead, a soft constraint is implicitly imposed by the posterior probabilities of the mixture components.

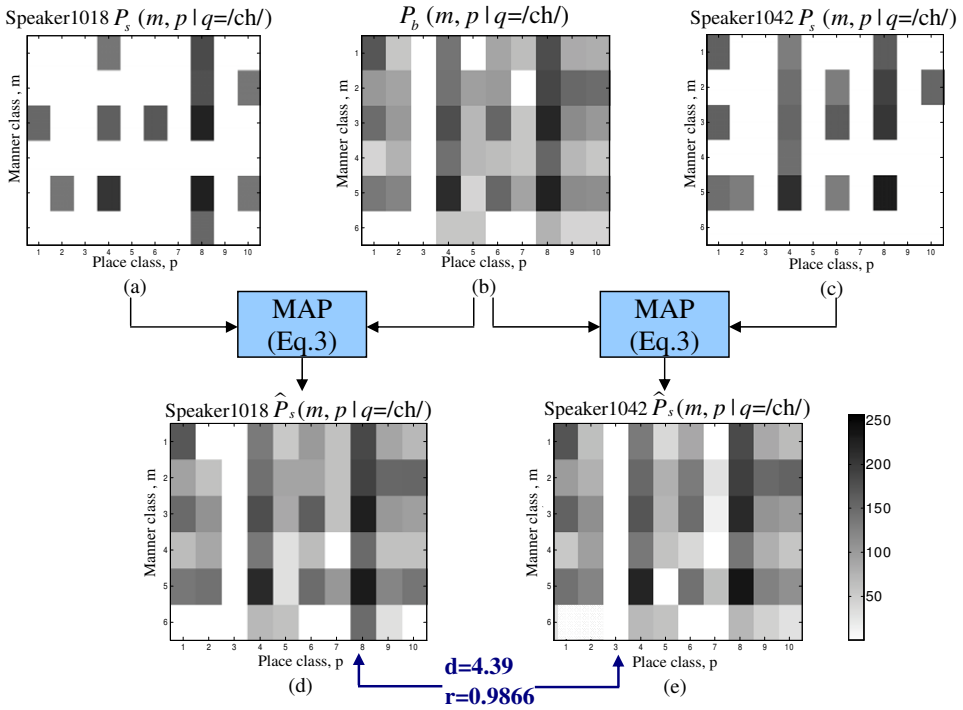


Figure 5.6: Phoneme-dependent AFCPMs correspond to phoneme /ch/ of (a) speaker 1018 from NIST00, (b) background speakers from NIST99, and (c) speaker 1042 from NIST00. (d) and (e): Phoneme-dependent speaker models of the two speakers adapted from (b) using the traditional MAP adaptation (see Method A in section 5.2.1). d and r represent the Euclidean distance and the correlation coefficient between the models pointed to by arrows. The 60 discrete probabilities corresponding to the combinations of the 6 manner and 10 place classes are nonlinearly quantized to 256 gray levels using log-scale, where white represents 0 and black represents 1.

5.2.2 New Adaptation Methods for AFCPMs

Our new adaptation methods attempt to utilize all of the available information. To relax the constrain imposed by classical MAP adaptation (see Problem 3 above), we introduce phoneme-independent models for target speakers and background speakers

as follows:

$$P_b(m,p|*) = P_b(L^{\mathrm{M}} = m, L^{\mathrm{P}} = p|\text{Background}) \tag{5.2}$$
$$= \frac{\#((m,p,*) \text{ in the data of all background speakers})}{\#((*,*,*) \text{ in the data of all background speakers})},$$
$$P_s(m,p|*) = P_s(L^{\mathrm{M}} = m, L^{\mathrm{P}} = p|\text{speaker} = s) \tag{5.3}$$
$$= \frac{\#((m,p,*) \text{ in the enrollment utterrence of speaker } s)}{\#((*,*,*) \text{ in the enrollment utterrence of speaker } s)},$$

where $m \in \mathcal{M}, p \in \mathcal{P}$ are defined in Section 3.1, and $(m,p,*)$ denotes the condition for which $L^{\mathrm{M}} = m, L^{\mathrm{P}} = p$. Based on the definition of $P_s(m,p|*), P_b(m,p|*)$ and $P_b(m,p|q)$, we can further derive:

$$\begin{aligned} P_s(m,p|*) &= \sum_{i=1}^{46} P_s(m,p|q_i)P_s(q_i), \\ P_b(m,p|q) &= \sum_{k=1}^{M} P_{s_k}(m,p|q)P(s_k), \\ P_b(m,p|*) &= \sum_{i=1}^{46} P_b(m,p|q_i)P_b(q_i), \end{aligned} \tag{5.4}$$

where M is the total number of background speakers used for training the background models, s_k is one of these background speakers, and q_i represents one of the 46 phonemes in English. The proof of the derivation is shown in Appendix B. Figure 5.7 illustrates how the phoneme-independent models are used for creating speaker models, which will be discussed next.

Method B:

Instead of adapting from the phoneme-dependent UBM, we can create the speaker model $\widehat{P}_s(m,p|q)$ by adapting the phoneme-independent speaker model $P_s(m,p|*)$, i.e.,

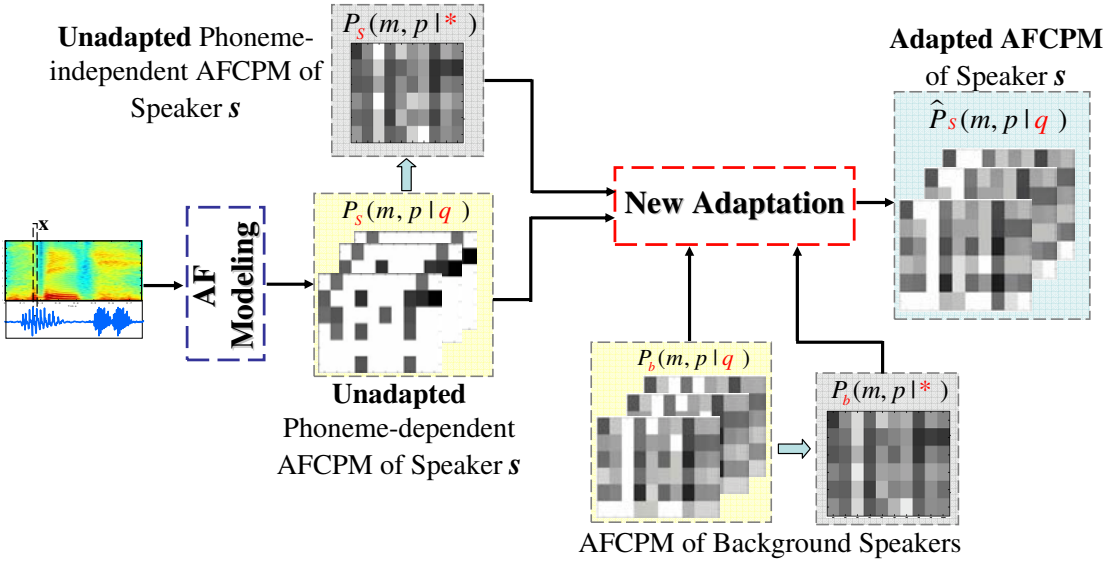


Figure 5.7: The information utilized for creating an adapted speaker model using the proposed adaptation method.

$$\widehat{P}_s(m,p|q) = \beta_s^q P_s(m,p|q) + (1-\beta_s^q) P_s(m,p|*). \tag{5.5}$$

Figure 5.5 illustrates the relationship (based on real data) between the unadapted and adapted speaker models created by this method. While this method can help solve Problems 1 and 3 mentioned in Section 5.2.1, it does have its own problem. The problem is that for a particular client, all of his/her phoneme-dependent models are adapted from the same phoneme-independent model, causing loss of phoneme-dependence in the client model. In fact, the method uses enrollment data only, as illustrated in Figure 5.3. This loss of phoneme-dependence, however, violates the requirement of the scoring procedure (see Section 4.2.5) where the speaker and background models are assumed to be phoneme-dependent. Fortunately, the phoneme-dependence in the client models can be easily retained by introducing a phoneme-dependent scaling factor in the adaption equation. This is to be discussed next.

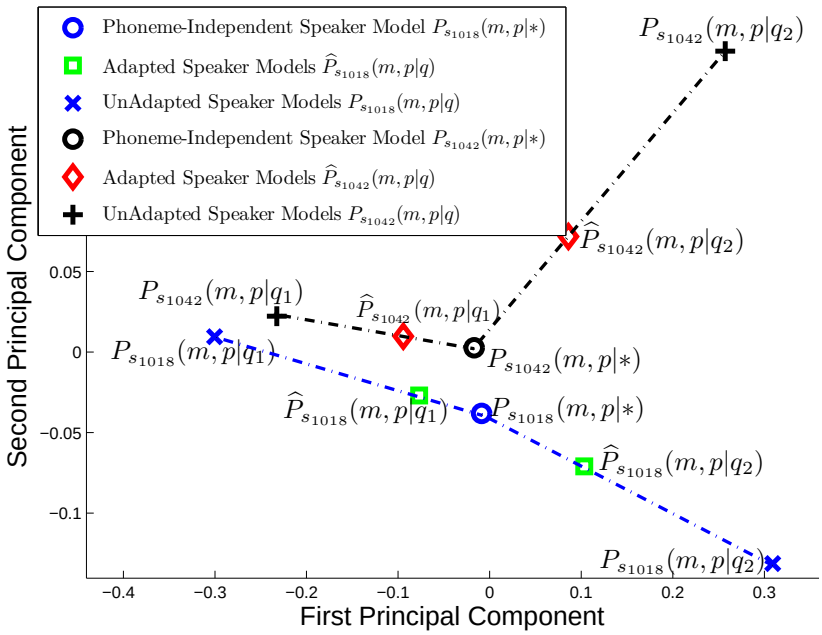


Figure 5.8: *Method B.* Relationship between the phoneme-independent speaker model, unadapted speaker models, and adapted speaker models for speakers 1018 and 1042 (q_1=/jh/, q_2=/uw/).

Method C:

In this method, a phoneme-dependent scaling factor is added to the adaptation formula in Eq. 5.5:

$$\widehat{P}_s(m,p|q) = \beta_s^q P_s(m,p|q) + (1-\beta_s^q)\cdot\left[\frac{P_b(m,p|q)}{P_b(m,p|*)}\cdot P_s(m,p|*)\right] \tag{5.6}$$

where $P_b(m,p|*)$ represents the phoneme-independent background model and $\frac{P_b(m,p|q)}{P_b(m,p|*)}$ is the scaling factor. With this factor, the model to be adapted becomes $\frac{P_b(m,p|q)}{P_b(m,p|*)}P_s(m,p|*)$. Therefore, the resulting target model $\widehat{P}_s(m,p|q)$ is now adapted from a model with certain degree of phoneme-dependence instead of adapting from a purely phoneme-independent model ($P_s(m,p|*)$).

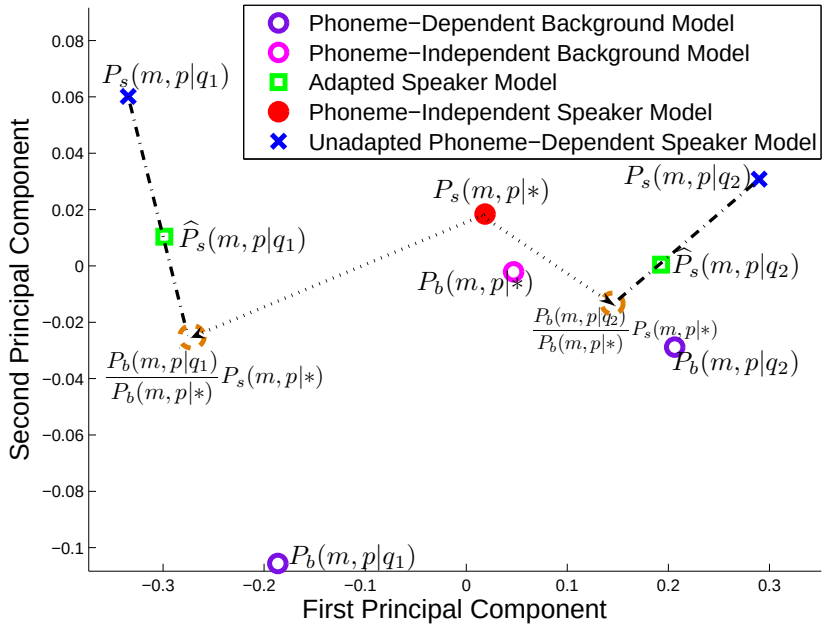


Figure 5.9: *Method C.* Relationship between the phoneme-independent speaker model, unadapted speaker models, and adapted speaker models for speaker 1018 (q_1=/jh/, q_2=/uw/).

Note that $\frac{P_b(m,p|q)}{P_b(m,p|*)}P_s(m,p|*)$ in Eq. 5.6 can also be written as $\frac{P_s(m,p|*)}{P_b(m,p|*)}P_b(m,p|q)$. In that case, we can interpret $\frac{P_s(m,p|*)}{P_b(m,p|*)}$ as a phoneme-independent scaling factor for the classical MAP adaptation in Eq. 5.1. This factor can help alleviates Problems 2 and 3 in classical MAP mentioned earlier, because it implicitly incorporates the speaker-dependent articulatory properties of other phonemes into the adaptation equation.

More interestingly, $\frac{P_b(m,p|q)}{P_b(m,p|*)}P_s(m,p|*)$ in Eq. 5.6 can be derived as (see the proof of this derivation in Appendix B):

$$\frac{P_b(m,p|q)}{P_b(m,p|*)}P_s(m,p|*) = \frac{\left[\sum_{k=1}^{M} P_{s_k}(m,p|q)P(s_k)\right] \cdot \left[\sum_{i=1}^{46} P_s(m,p|q_i)P_s(q_i)\right]}{\sum_{k=1}^{M}\sum_{i=1}^{46} P_{s_k}(m,p|q_i)P(s_k)P_b(q_i)} \tag{5.7}$$

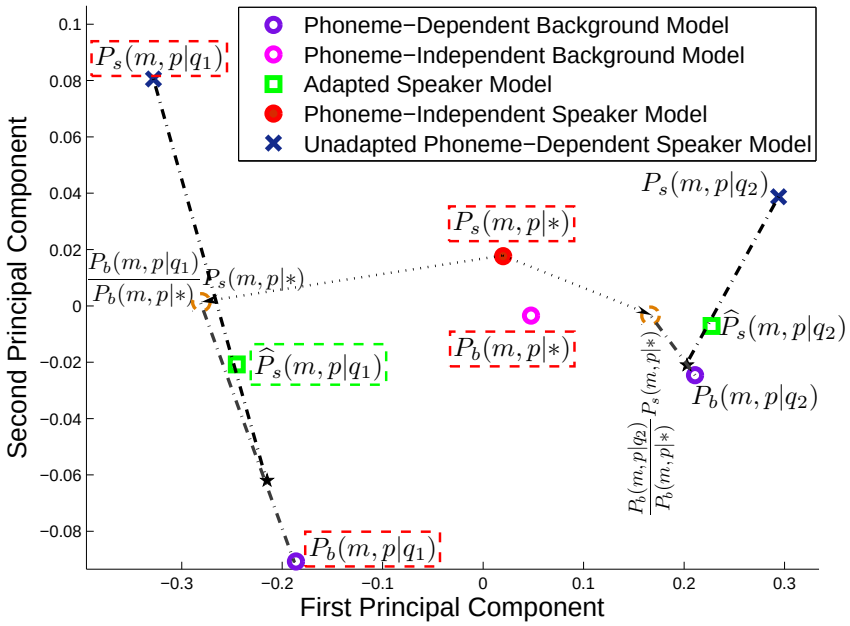


Figure 5.10: *Method D.* Relationship between the phoneme-independent speaker model, unadapted speaker models, and adapted speaker models for speaker 1018. (q_1=/jh/, q_2=/uw/ and the marker '★' represents the term inside the square brackets in Eq. 5.11.)

where M is the total number of background speakers used for training the background models, s_k is one of these background speakers, and q_i represents one of 46 phoneme in English. If we assume $P_b(q_1) = \cdots = P_b(q_i) = \cdots = P_s(q_i) =$ constant and $P(s_1) = P(s_2) = \cdots = P(s_k) =$ constant, then more clearly, we have:

$$\frac{P_b(m,p|q)}{P_b(m,p|*)}P_s(m,p|*) = \frac{\left[\sum_{k=1}^{M} P_{s_k}(m,p|q)\right] \cdot \left[\sum_{i=1}^{46} P_s(m,p|q_i)\right]}{\sum_{k=1}^{M}\sum_{i=1}^{46} P_{s_k}(m,p|q_i)}. \tag{5.8}$$

Eqs. 5.7 and 5.8 suggest that all of the available information have been utilized during the adaptation process. Figure 5.9 illustrates the projection of the unadapted and adapted speaker models on their first two principle components.

Method D:

It becomes clear that Method A is likely to impose too much constraint on the adaptation. Method B aims to relax such constraint by introducing a phoneme-independent model in its adaptation equation. However, the relaxation may be too far so that the phoneme-dependent scaling factor in Method C is necessary to limit the loss of phoneme-dependence. Nevertheless, the target models created by Method C depend implicitly on the phoneme-dependent background models $P_b(m, p|q)$ through the scaling factor. To strengthen the dependence of these background models while allowing certain degree of phoneme-independence, we may combine Methods A and C. We refer to the resulting adaptation as Method D whose adaptation equation is written as:

$$\widehat{P}_s(m, p|q) = \beta_s^q P_s(m, p|q) + (1 - \beta_s^q) \left[\alpha_b^q P_b(m, p|q) + (1 - \alpha_b^q) \frac{P_b(m, p|q)}{P_b(m, p|*)} P_s(m, p|*) \right] \tag{5.9}$$

where, $\alpha_b^q \in [0, 1]$ is a phoneme-dependent adaptation coefficient. It is obtained by

$$\alpha_b^q = \frac{\#((*, *, q) \text{ in the utterances of all background speakers})}{\#((*, *, q) \text{ in the utterances of all background speakers}) + r_\alpha} \tag{5.10}$$

where r_α is also a fixed relevance factor.

Fig. 5.10 illustrates the relationship between different models in Method D, and Fig. 5.11 explains why this method is better than Method A via an illustrative example. Comparing Figs. 5.6 and 5.11 reveals that the Euclidean distance and dissimilarity between the AFCPM models of speakers 1018 and 1042 become larger (the distance increases from 4.39 to 14.17 and the correlation coefficient reduces from 0.9966 to 0.8013). Therefore Method D makes the speaker models better in discriminating speakers.

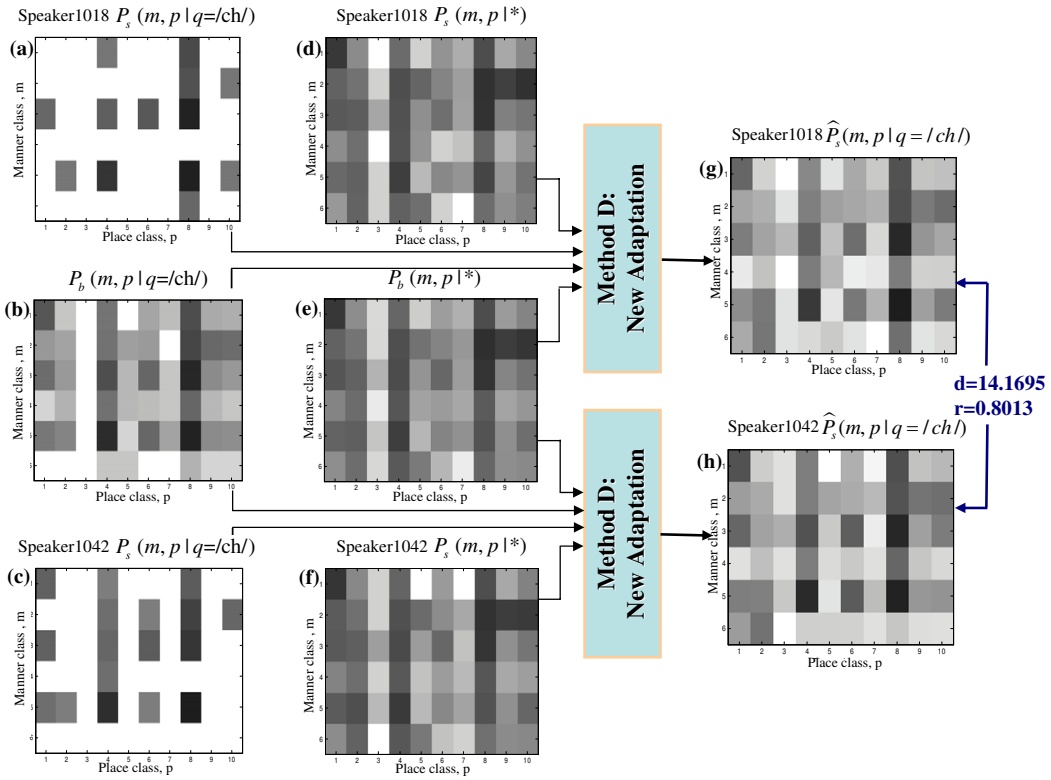


Figure 5.11: Phoneme-dependent AFCPMs ((g) and (h)) of speakers 1018 and 1042 created by Method D. (a) and (c): Unadapted speaker models. (b) Phoneme-dependent background model. (d) and (f): Phoneme-independent speaker models. (e) Phoneme-independent background model. d and r represent the Euclidean distance and the correlation coefficient between the adapted models pointed to by arrows.

Method E:

Using the same idea in Method D, here we mix phoneme-independent speaker model and phoneme-dependent UBMs which is adjusted by another scaling factor to adapt the original speaker model (Eq. 3.2). The method is described mathematically as follows:

$$\widehat{P}_s(m,p|q) \;=\; \beta_s^q P_s(m,p|q) \;+\; (1 \;-\; \beta_s^q)\left[\alpha_b^q P_b(m,p|q)\frac{P_s(m,p|*)}{P_b(m,p|*)} + (1-\alpha_b^q)P_s(m,p|*)\right] \tag{5.11}$$

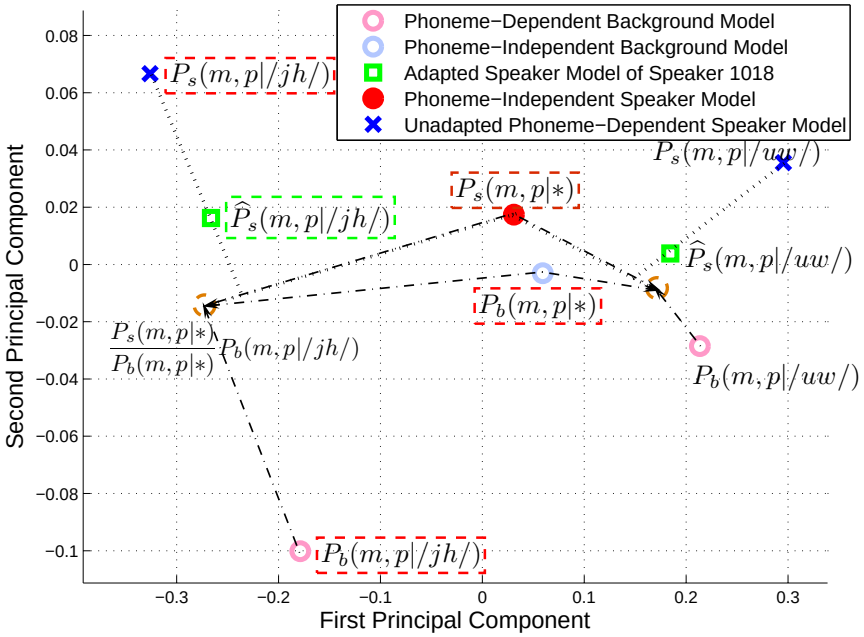


Figure 5.12: *Method E.* Relationship between the unadapted, adapted phoneme-dependent and phoneme-independent speaker models for speaker 1018 /jh/, /uw/ and corresponding phoneme-dependent and phoneme-independent background models in Method E.

where $\frac{P_s(m,p|*)}{P_b(m,p|*)}$ is a phoneme-independent scaling factor used to change the component values of phoneme-dependent UBMs from a general background-speaker level to a speaker-specific level. The adaptation procedure of method D is illustrated in Figure 5.12.

5.3 Scoring Based on Adapted Models

The scoring method is identical to the one in Section 4.2.5. However, this time the speaker models $\widehat{P}_s(m,p|q)$ created by using different adaptation methods discussed in Section 5.2 are used instead.

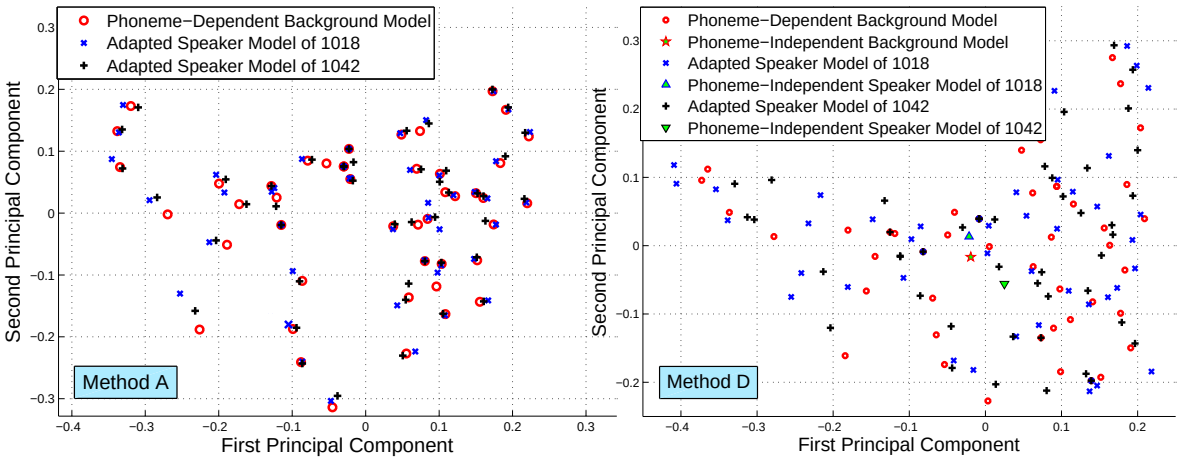


Figure 5.13: The distribution of all adapted phoneme-dependent speaker models and phoneme-dependent background models in principal component space for speaker 1018 and 1042 based on Method A (left) and Method D (right).

5.4 Experiments and Results

5.4.1 Procedures

The procedure and data are identical the those in in Section 3.4, except that phoneme-dependent speaker models were created using Methods A to E.

5.4.2 Results and Discussion

Figure 5.13 shows the relationship between the phoneme-dependent background and adapted models (corresponding to 46 phonemes) of two speakers for Methods A and D. Apparently, Problem 1 in Method A (left figure) mentioned in Section 5.2.1 does not appear in Method D (right figure).

Table 5.1 shows the equal error rate (EER) and p-values [48] (with respect to Method A) achieved by different adaptation methods. It shows that Methods C, D, and E achieve a lower error rate as compared to the classical MAP adaption. This confirms our earlier argument that better speaker models can be obtained by adapting

the phoneme-independent models in addition to the phoneme-dependent models.

Adaptation Method	**EER (%)**	***p*-values**
Method A	26.34	—
Method B	26.81	0.04560
Method C	25.68	0.00008
Method D	24.88	0.00000
Method E	25.58	0.00006

Table 5.1: EERs obtained by phoneme-dependent AFCPMs created by MAP-based adaptation methods described in Section 5.2. The p-values between the classical MAP and the new adaptation methods are listed in the last column.

The DET plots corresponding to Table 5.1 are shown in Figure 5.14. Evidently, Method D achieves the best performance across a wide range of decision threshold. It was found that the proposed adaptation approaches can effectively solve the data sparseness problem, resulting in a significantly lower error rate. Apparently, Problems 2 and 3 in Method A have also been overcome by Method D. The fusion DET plots shown in Figure 5.15 also demonstrate that the AFCPMs created by Method D are complementary to the acoustic GMMs, leading to a slightly better performance when the scores of the two types of models are combined (compare GMM+AFCPM(Method D) with GMM+AFCPM (MAP)).

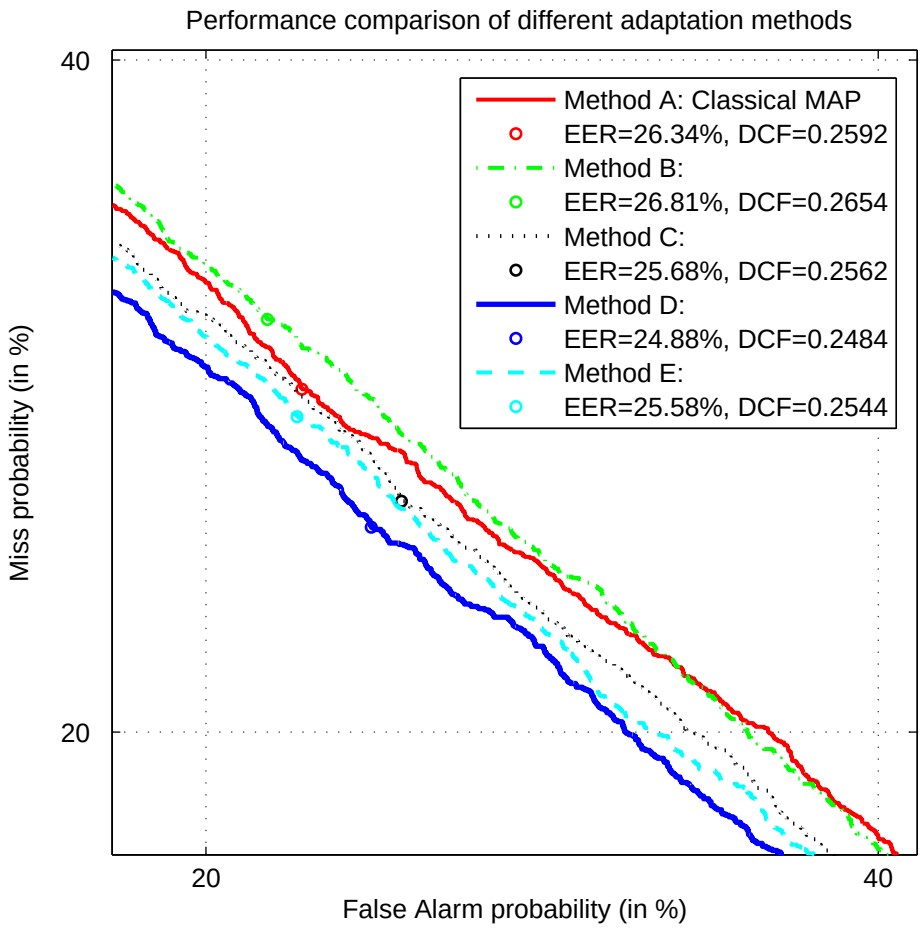


Figure 5.14: DET performance of AFCPM speaker verification system using different adaptation methods.

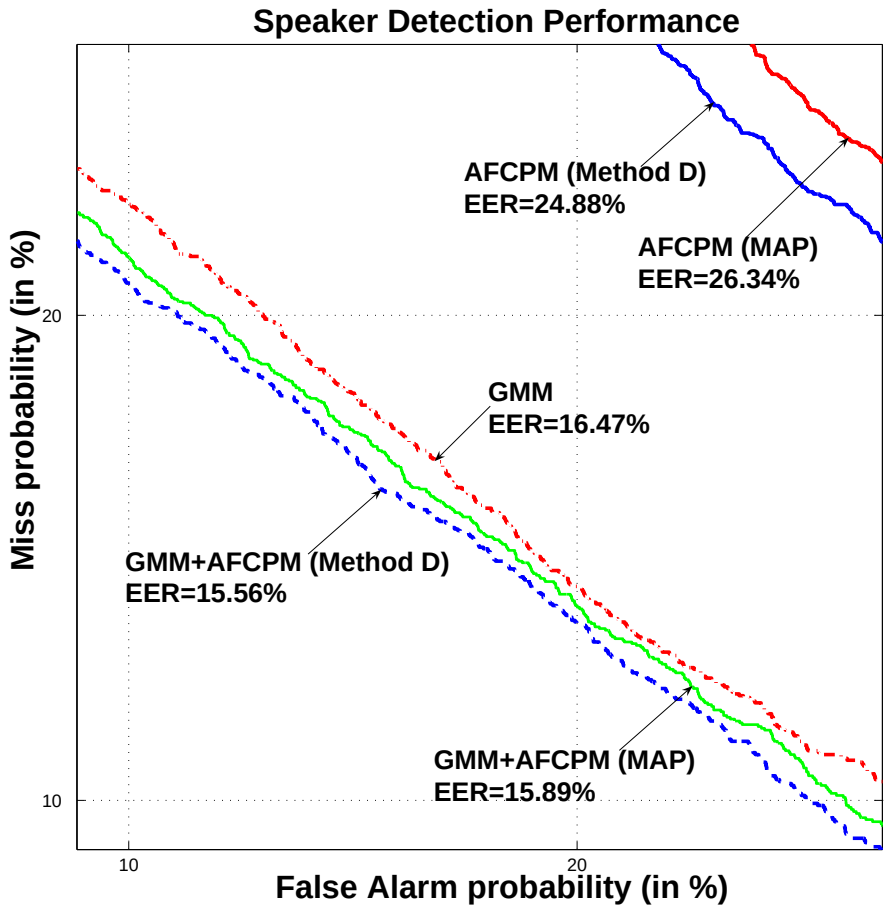


Figure 5.15: DET performance of AFCPM (MAP), AFCPM (Method D), GMM and their fusions.

Chapter 6

ARTICULATORY-FEATURE BASED SEQUENCE KERNEL FOR HIGH-LEVEL SPEAKER VERIFICATION

In GMM-UBM, CD-AFCPM, and PW-CD-AFCPM, scoring is done at the frame-level, i.e., each frame of speech is scored separately and then frame-based scores are accumulated to produce the final utterance-based score for classification. This frame-based scoring scheme has two drawbacks. First, treating the frames individually may not be able to fully capture the sequence information contained in the utterance. Second, the goal of speaker verification is to minimize classification errors on test utterances rather than on individual speech frames.

These drawback motivates us to derive a sequence-based approach in which an utterance is considered as comprising of a sequence of symbols and the utterance-based score can be obtained from an SVM [56] through a kernel function of the sequence of symbols. SVM can produce complex decision function regions without a large amount of training data. However, SVMs are normally only able to classify data of fixed dimensionality whereas speech utterances are typically parameterized as variable length sequences of observation vectors. This has led to the use sequence kernels. Sequence kernels implicitly map variable length observation sequences into a fixed-dimensional vector typically via generative models (GMMs). Many studies [57–59] have been shown that applying sequence-kernel based SVM to short-term spectra-based generative models can achieve promising results.

A critical issue in using SVM is the design of kernels. Because there is no universal kernel that is suitable for all problems, it is imperative to derive a special sequence

kernel for AFCPM. This chapter derives an articulatory feature-based kernel for high-level speaker verification. The relationship between traditional frame-based AFCPM scoring and the utterance-level kernel-based scoring is discussed.

6.1 Motivation

Research has shown that the performance of pattern classification systems can be improved by training the classifiers discriminatively via supervised learning. A well-known example of discriminative training for speech and speaker recognition is the minimum classification error (MCE) training [60–62]. The drawback of MCE, however, is that it optimizes the objective function via gradient descent, which usually takes longer training time and is less stable as compared to the maximum-likelihood approach. Another approach to incorporating class information into the learning stage is to consider the speaker and background models as high-dimensional supervectors. SVMs are then discriminatively trained to classify the supervectors in the high-dimensional space [57, 58] This chapter focuses on the latter approach.

6.2 Phonetic-Class Dependent AFCPM Supervectors

Figure 6.1 shows the verification phase of a high-level speaker verification system that uses AF-kernels. The first step is to create the CD-AFCPM (see Chapter 3); then, for each target speaker, G speaker models (each model has 60 values) are vectorized and concatenated to form a single supervector, called CD-AFCPM Supervector.[1] The process maps a test utterance to a point in $60G$-dimensional vector space. The procedure of CD-AFCPM supervector extraction is illustrated in Figure 6.2.

[1]The procedure is also applicable to PD-AFCPM with $G = 46$. For clarity, we focus on CD-AFCPM in the sequel.

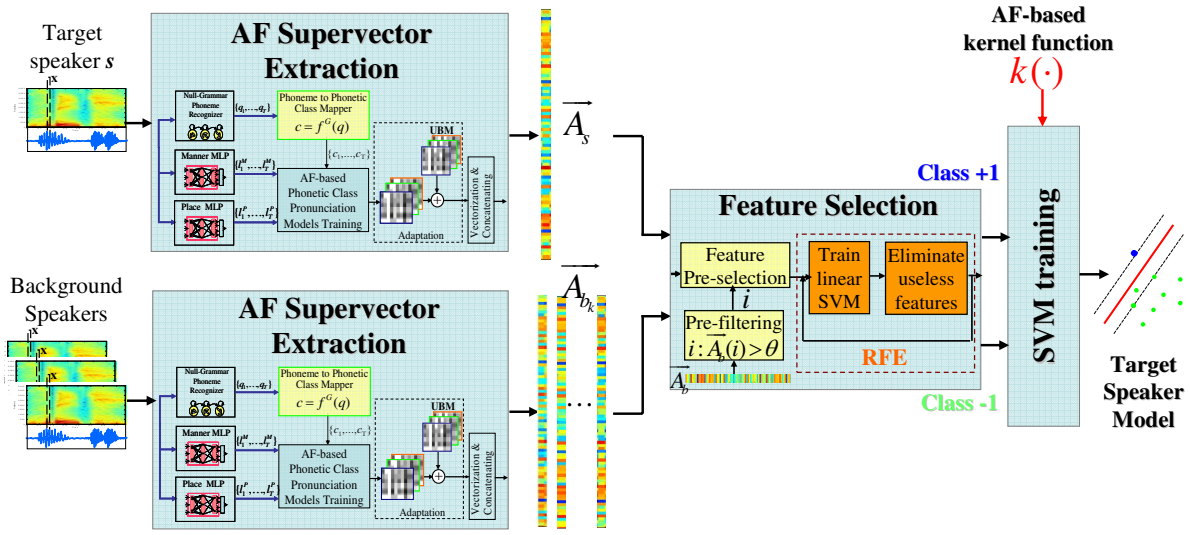


Figure 6.1: The training procedure of the AF kernel-based high-level speaker verification system.

6.3 Feature Selection

The dimensionality of CD-AFCPM and PD-AFCPM is 720 (when $G = 12$) and 2760, respectively. Many of these features, however, may be redundant or having low discriminative power. Therefore, extracting the relevant features from the high-dimensional supervectors is expected to improve verification performance.

We applied the recursive feature elimination (RFE) algorithm [63] for the feature selection. The procedure is illustrated in Figure 6.3. The feature selection process is divided into two steps. In Step 1, irrelevant features are weeded out using a pre-filtering approach. More precisely, the background CD-AFCPMs are vectorized and feature elements with value smaller than a threshold are removed. This step avoids the numerical difficult that may occur when the CD-AFCPMs supervectors are normalized during the evaluation of the kernel function (see Eqs. 6.10 and 6.11). Then, in Step 2, based on the remaining features in Step 1, a CD-AFCPM supervector is constructed for each target speaker, and 618 CD-AFCPM supervectors are constructed

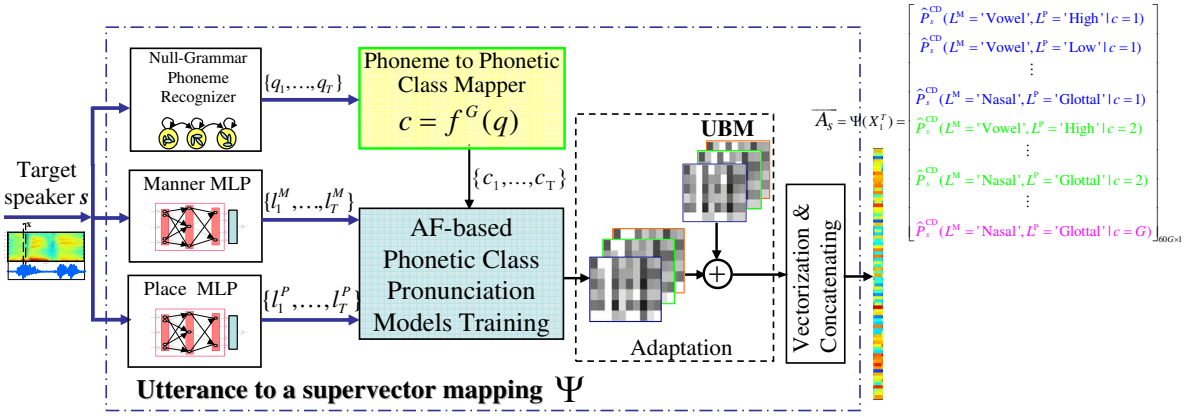


Figure 6.2: The procedure of extracting phonetic-class AFCPM supervectors for AF kernel-based high-level speaker verification.

from 618 background speakers. Then, for each speaker, feature elimination is done by applying SVM-RFE [63] to the dataset formed by the speaker's CD-AFCPM supervector (positive class) and background speakers' CD-AFCPM supervectors (negative class). Note that Step 2 is applied to each of the target speakers, meaning that each speaker has their own feature set.

The effect of feature selection is shown in Figure 6.4. In the figure, each column corresponds to a portion of a speaker's CD-AFCPM supervector. A row with small variation (almost identical color intensity) suggests that the corresponding feature is not speaker dependent and therefore can be removed without sacrifying classification performance. We can see from Figure 6.4 (right panel) that features of low discriminative power have been eliminated.

6.4 Articulatory Feature-Based Kernels

6.4.1 Another Interpretation of Articulatory Feature-Based LR Scoring

Given a test utterance $X_1^T = \{X_1, \ldots, X_t, \ldots, X_T\}$, speaker models $\widehat{P}_s^{\mathrm{CD}}(m, p|c)$, and UBMs $P_b^{\mathrm{CD}}(m, p|c)$, we can express the frame-based likelihood-ratio (LR) score as

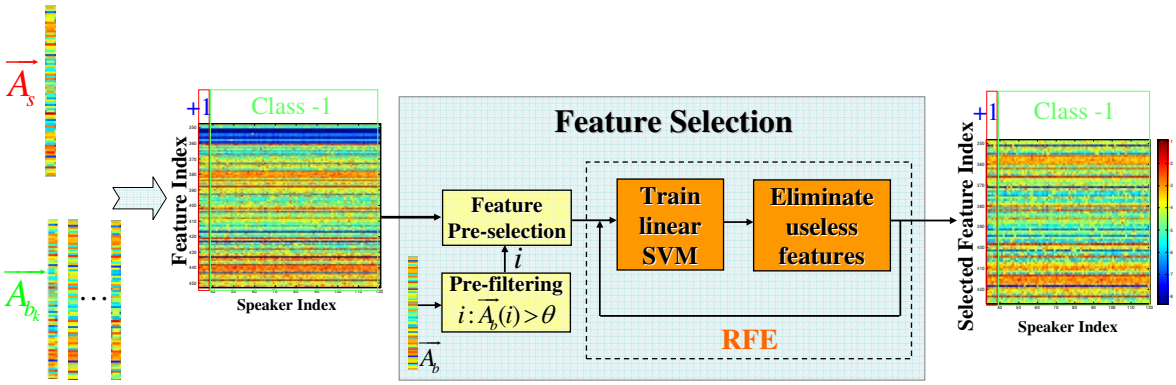


Figure 6.3: The procedure of selecting relevant features from CD-AFCPM supervectors. Columns represent speakers and rows represent features.

follows:

$$S_{\text{CD-AFCPM}}(X_1^T) = \frac{1}{T}\sum_{t=1}^{T}\left(\log\frac{\widehat{p}_s^{\text{CD}}(X_t)}{p_b^{\text{CD}}(X_t)}\right) = \sum_{c=1}^{G}\left(\frac{1}{T}\sum_{t:f^G(q_t)=c}\left(\log\frac{\widehat{p}_s^{\text{CD}}(X_t)}{p_b^{\text{CD}}(X_t)}\right)\right),$$

where $f^G(q_t)$ is one of the mapping function mentioned in Section 3.3.1.

Based on Eqs. 3.13 and 3.14, we can further express the LR score as:

$$\begin{aligned} S_{\text{CD-AFCPM}}(X_1^T) &= \sum_{c=1}^{G}\left(\frac{1}{T}\sum_{t:f^G(q_t)=c}\left(\log\frac{\widehat{P}_s^{\text{CD}}(l_t^{\text{M}}, l_t^{\text{P}}|c)}{P_b^{\text{CD}}(l_t^{\text{M}}, l_t^{\text{P}}|c)}\right)\right) \\ &= \sum_{c=1}^{G}\frac{1}{T}\left(\sum_{m\in\mathcal{M}}\sum_{p\in\mathcal{P}}\left(\sum_{t:\begin{cases} f^G(q_t)=c \\ l_t^{\text{M}}=m, l_t^{\text{P}}=p \end{cases}}\left(\log\frac{\widehat{P}_s^{\text{CD}}(l_t^{\text{M}}=m, l_t^{\text{P}}=p|c)}{P_b^{\text{CD}}(l_t^{\text{M}}=m, l_t^{\text{P}}=p|c)}\right)\right)\right) \end{aligned} \tag{6.1}$$

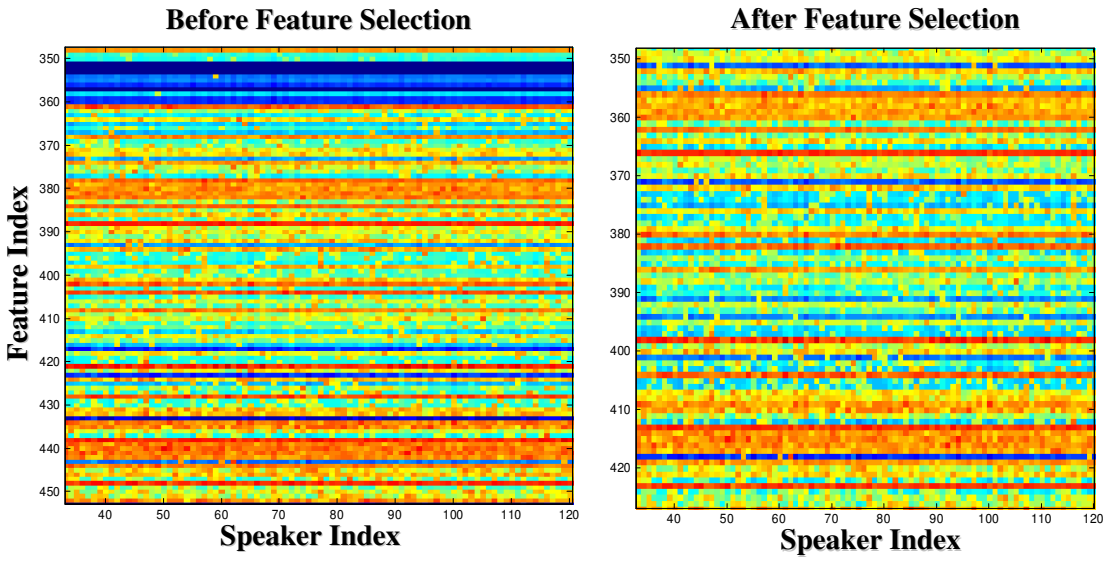


Figure 6.4: Effect of feature selection on CD-AFCPM supervectors. A row with small variation (almost identical color intensity) suggests that the corresponding feature is not speaker dependent and therefore can be removed without scarifying classification accuracy.

$$
\begin{aligned}
&= \sum_{c=1}^{G} \frac{T_c}{T} \left(\frac{1}{T_c} \sum_{i=1}^{60} \left(\left(\log \frac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_i|c)}{P_b^{\mathrm{CD}}(\mathcal{L}_i|c)} \right) \sum_{t:\begin{cases} f^G(q_t) = c \\ \mathcal{L}_i \end{cases}} 1 \right) \right) \\
&= \sum_{c=1}^{G} \frac{T_c}{T} \left(\frac{1}{T_c} \sum_{i=1}^{60} \left(\left(\log \frac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_i|c)}{P_b^{\mathrm{CD}}(\mathcal{L}_i|c)} \right) N_{i,c} \right) \right) \\
&= \sum_{c=1}^{G} \frac{T_c}{T} \left(\sum_{i=1}^{60} \left(\left(\log \frac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_i|c)}{P_b^{\mathrm{CD}}(\mathcal{L}_i|c)} \right) \frac{N_{i,c}}{T_c} \right) \right)
\end{aligned}
$$

where

$$\mathcal{L}_1 = \{l_t^{\mathrm{M}} = \text{‘Vowel’}, l_t^{\mathrm{P}} = \text{‘High’ for any } t\},$$

$$\cdots$$

$$\mathcal{L}_{60} = \{l_t^{\mathrm{M}} = \text{‘Lateral’}, l_t^{\mathrm{P}} = \text{‘Glottal’ for any } t\},$$

$N_{i,c}$ is the number of frames that belong to phonetic class c and $\mathcal{L}_i$, and T_c is the number of frames that belong to phonetic class c.

Assume that the test utterance is produced by a claimant claiming a speaker identity s. Then, we can obtain the CD-AFCPM of the claimant as follows:

$$\begin{aligned}
&P^{\mathrm{CD}}_{claimant}(m,p|c) \\
&= P^{\mathrm{CD}}_{claimant}(L^{\mathrm{M}} = m, L^{\mathrm{P}} = p|\text{PhoneClass} = c, \text{Speaker} = claimant) \\
&= P^{\mathrm{CD}}_{claimant}(\mathcal{L}_i|c) \\
&= \frac{\#((m,p,c) \text{ in the utterances of the claimant})}{\#((*,*,c) \text{ in the utterances of the claimant})} \\
&= \frac{N_{i,c}}{T_c},
\end{aligned} \tag{6.2}$$

where index i corresponds to the i-th combination of the manner and place class (m,p).

Substituting Eq. 6.2 in Eq. 6.1, we obtain:

$$\begin{aligned}
S_{\text{CD-AFCPM}}(X_1^T) &= \sum_{c=1}^{G} \frac{T_c}{T} \left(\sum_{i=1}^{60} \left(\left(\log \frac{\widehat{P}^{\mathrm{CD}}_s(\mathcal{L}_i|c)}{P^{\mathrm{CD}}_b(\mathcal{L}_i|c)} \right) P^{\mathrm{CD}}_{claimant}(\mathcal{L}_i|c) \right) \right) \\
&= \sum_{c=1}^{G} \left\langle \begin{bmatrix} \log \frac{\widehat{P}^{\mathrm{CD}}_s(\mathcal{L}_1|c)}{P^{\mathrm{CD}}_b(\mathcal{L}_1|c)} \\ \log \frac{\widehat{P}^{\mathrm{CD}}_s(\mathcal{L}_2|c)}{P^{\mathrm{CD}}_b(\mathcal{L}_2|c)} \\ \cdots \\ \log \frac{\widehat{P}^{\mathrm{CD}}_s(\mathcal{L}_{60}|c)}{P^{\mathrm{CD}}_b(\mathcal{L}_{60}|c)} \end{bmatrix}_{60}, \begin{bmatrix} \frac{T_c}{T} P^{\mathrm{CD}}_{claimant}(\mathcal{L}_1|c) \\ \frac{T_c}{T} P^{\mathrm{CD}}_{claimant}(\mathcal{L}_2|c) \\ \cdots \\ \frac{T_c}{T} P^{\mathrm{CD}}_{claimant}(\mathcal{L}_{60}|c) \end{bmatrix}_{60} \right\rangle
\end{aligned} \tag{6.3}$$

$$= \left\langle \begin{bmatrix} \log \dfrac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_1|c=1)}{P_b^{\mathrm{CD}}(\mathcal{L}_1|c=1)} \\ \cdots \\ \log \dfrac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_{60}|c=1)}{P_b^{\mathrm{CD}}(\mathcal{L}_{60}|c=1)} \\ \log \dfrac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_1|c=2)}{P_b^{\mathrm{CD}}(\mathcal{L}_1|c=2)} \\ \cdots \\ \log \dfrac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_{60}|c=2)}{P_b^{\mathrm{CD}}(\mathcal{L}_{60}|c=2)} \\ \cdots \\ \cdots \\ \log \dfrac{\widehat{P}_s^{\mathrm{CD}}(\mathcal{L}_{60}|c=12)}{P_b^{\mathrm{CD}}(\mathcal{L}_{60}|c=12)} \end{bmatrix}_{720}, \begin{bmatrix} w_1 P_{claimant}^{\mathrm{CD}}(\mathcal{L}_1|c=1) \\ \cdots \\ w_1 P_{claimant}^{\mathrm{CD}}(\mathcal{L}_{60}|c=1) \\ w_2 P_{claimant}^{\mathrm{CD}}(\mathcal{L}_1|c=2) \\ \cdots \\ w_2 P_{claimant}^{\mathrm{CD}}(\mathcal{L}_{60}|c=2) \\ \cdots \\ \cdots \\ w_{12} P_{claimant}^{\mathrm{CD}}(\mathcal{L}_{60}|c=12) \end{bmatrix}_{720} \right\rangle, (\because G = 12)$$

where $w_i = \dfrac{T_i}{T}$ $(i = 1, \ldots, G)$ and $G = 12$. As a result, we have

$$\begin{aligned} S_{\text{CD-AFCPM}}(X_1^T) &= \left\langle \log \frac{\overrightarrow{A_s}}{\overrightarrow{A_b}}, \overrightarrow{A_c'} \right\rangle \\ &= \left\langle \log \frac{\overrightarrow{A_s}}{\overrightarrow{A_b}}, \overrightarrow{w} .* \overrightarrow{A_c} \right\rangle \end{aligned} \tag{6.4}$$

where $\overrightarrow{A_s}, \overrightarrow{A_b}$ and $\overrightarrow{A_c}$ stand for the AF supervector of the speaker, background and claimant, respectively, $\log \dfrac{\overrightarrow{X}}{\overrightarrow{Y}} \equiv \left[\log \dfrac{x_1}{y_1}, \ldots, \log \dfrac{x_N}{y_N} \right]^{\mathrm{T}}$ and $\overrightarrow{X} .* \overrightarrow{Y} \equiv \left[x_1 y_1, \cdots, x_N y_N \right]^{\mathrm{T}}$, where x_i and y_i are elements of $\overrightarrow{X}$ and $\overrightarrow{Y}$, respectively. We can see from Eq. 6.4 that the traditional frame-based LR scoring for discrete models can be computed in another way: compute the dot product between a speaker-dependent supervector derived from the models of the target speakers and a weighted supervector obtained from the claimant's model.

Further, denote $\overrightarrow{A_c'}$ as the weighted AF supervector of the claimant, we have

$$
\begin{aligned}
S_{\text{CD-AFCPM}}(X_1^T) &= \frac{1}{T}\sum_{t=1}^{T}\left(\log\frac{\widehat{p}_s^{\,\text{CD}}(X_t)}{p_b^{\text{CD}}(X_t)}\right) \\
&= \left\langle \log\frac{\overrightarrow{A_s}}{\overrightarrow{A_b}}, \overrightarrow{A_c'} \right\rangle \\
&= \left\langle \overrightarrow{A_c'}, \log\overrightarrow{A_s} \right\rangle - \left\langle \overrightarrow{A_c'}, \log\overrightarrow{A_b} \right\rangle .
\end{aligned}
\tag{6.5}
$$

Eq. 6.5 suggests an alternative approach to implementing the traditional LR scoring. This is shown in Figure 6.5. We can see from Figure 6.5 that if we can replace

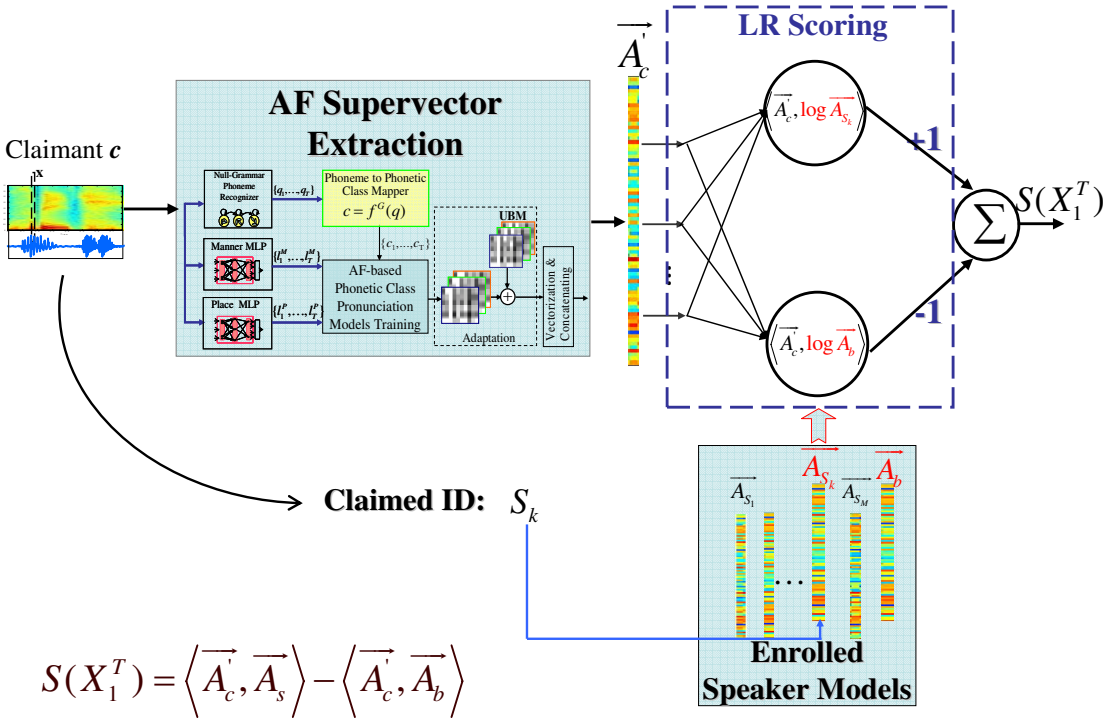


Figure 6.5: An alternative implementation of the traditional log-likelihood scoring in CD-AFCPM speaker verification.

the fixed '+1' and '−1' multiplication factors in the LR scoring block by varying weights α_i, the result may probably be improved. These weights can be optimally

determined by SVM training. In order to apply SVM and to make sure that the training algorithm converges to a stable solution, the function inside the processing nodes ("circle") in Figure 6.5 should satisfy the Mercer condition [56]. Unfortunately, $f(\overrightarrow{X}, \overrightarrow{Y}) = \left\langle \overrightarrow{X}, \log \overrightarrow{Y} \right\rangle$ does not satisfy the Mercer condition because $\left\langle \overrightarrow{X}, \log \overrightarrow{Y} \right\rangle$ cannot be written as $\left\langle \Phi(\overrightarrow{X}), \Phi(\overrightarrow{Y}) \right\rangle$, and thus cannot be used as a kernel function. Therefore, we derive an AF kernel function that satisfies the Mercer condition in the next section.

6.4.2 Deriving Kernels from Similarity Scores

Given AF-based supervectors $\overrightarrow{A}_s$ and $\overrightarrow{A}_b$ obtained by mapping the speech utterances of speaker s and background speakers into a fixed-dimension input space, the similarity score between the model deriving from the test utterance X_1^T of claimant c and the model of speaker s can be computed by a similarity function (discriminant function):

$$\begin{aligned} Similarity(\overrightarrow{A}_c, \overrightarrow{A}_s) &= \frac{1}{T}\sum_{t=1}^{T}\left(\log\frac{\widehat{p}_s^{\mathrm{CD}}(X_t)}{p_b^{\mathrm{CD}}(X_t)}\right) \\ &= \left\langle \overrightarrow{A_c'}, \log\frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \right\rangle \end{aligned} \tag{6.6}$$

where Eq. 6.5 has been used in the derivation.

Our goal is to make Eq. 6.6 symmetric and satisfy the Mercer condition. To this end, we expand $\log(x)$ at $x = 1$ as a Taylor series:

$$\begin{aligned} \log(x) &= \sum_{n=0}^{\infty}\frac{\log^{(n)}(1)}{n!}(x-1)^n \\ &= (x-1) - \frac{1}{2}(x-1)^2 + \frac{1}{3}(x-1)^3 + \mathcal{O}\left((x-1)^4\right). \end{aligned} \tag{6.7}$$

Because the speaker models are adapted from the UBMs, $\dfrac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \to \overrightarrow{1}$. Therefore, we

can ignore the high orders of $\left(\frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} - \overrightarrow{1}\right)$ and approximate Eq. 6.6 as:

$$\begin{aligned} Similarity(\overrightarrow{A}_c, \overrightarrow{A}_s) &= \frac{1}{T}\sum_{t=1}^{T}\left(\log\frac{\widehat{p}_s^{\mathrm{CD}}(X_t)}{p_b^{\mathrm{CD}}(X_t)}\right) \\ &= \left\langle \overrightarrow{A_c'}, \log\frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \right\rangle \\ &\approx \left\langle \overrightarrow{A_c'}, \left(\frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} - \overrightarrow{1}\right) \right\rangle = \left\langle \overrightarrow{A_c'}, \frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \right\rangle - \left\langle \overrightarrow{A_c'}, \overrightarrow{1} \right\rangle \\ &= \left\langle \overrightarrow{A_c'}, \frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \right\rangle - G. \end{aligned} \tag{6.8}$$

Because the number of phonetic classes (G) is a constant for every speaker, it can be dropped without affecting the classification:

$$\begin{aligned} Similarity(\overrightarrow{A}_c, \overrightarrow{A}_s) &\approx \left\langle \overrightarrow{A_c'}, \frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \right\rangle = \left\langle \frac{\overrightarrow{A_c'}}{\sqrt{\overrightarrow{A_b}}}, \frac{\overrightarrow{A_s}}{\sqrt{\overrightarrow{A_b}}} \right\rangle \\ &= \left\langle \frac{\overrightarrow{w}.*\overrightarrow{A_c}}{\sqrt{\overrightarrow{A_b}}}, \frac{\overrightarrow{A_s}}{\sqrt{\overrightarrow{A_b}}} \right\rangle \\ &\approx \left\langle \frac{\sqrt{\overrightarrow{w_b}}.*\overrightarrow{A_c}}{\sqrt{\overrightarrow{A_b}}}, \frac{\sqrt{\overrightarrow{w_b}}.*\overrightarrow{A_s}}{\sqrt{\overrightarrow{A_b}}} \right\rangle \end{aligned} \tag{6.9}$$

where $\overrightarrow{w_b} = \left[\overbrace{\frac{T_1^b}{T},\cdots,\frac{T_1^b}{T}}^{60}, \overbrace{\frac{T_2^b}{T},\cdots,\frac{T_2^b}{T}}^{60},\cdots,\overbrace{\frac{T_G^b}{T},\cdots,\frac{T_G^b}{T}}^{60},\right]_{60G}^{\mathrm{T}}$ contains the phonetic-class weights obtained from the UBMs, which is used to approximate $\overrightarrow{w}$ in Eq. 6.4. The approximation aims to make the similarity measure symmetric.

Finally, we write the similarity function (Eq. 6.9) as kernel:

$$K_{\mathrm{AF}}(\overrightarrow{A}_c, \overrightarrow{A}_s) = \left\langle \frac{\sqrt{\overrightarrow{w_b}} * \overrightarrow{A_c}}{\sqrt{\overrightarrow{A_b}}}, \frac{\sqrt{\overrightarrow{w_b}} * \overrightarrow{A_s}}{\sqrt{\overrightarrow{A_b}}} \right\rangle = \left\langle \varphi(\overrightarrow{A_c}), \varphi(\overrightarrow{A_s}) \right\rangle \tag{6.10}$$

where the mapping $\varphi(\cdot)$ is defined as:

$$\varphi(\overrightarrow{X}) = \frac{\sqrt{\overrightarrow{w_b}} * \overrightarrow{X}}{\sqrt{\overrightarrow{A_b}}}. \tag{6.11}$$

Note that the kernel in Eq. 6.10 depends on the models that we used to represent the target speaker. Therefore, if we vectorise a variable-length observation sequence $X_1^T = \{X_1, \ldots, X_t, \ldots, X_T\}$ from the speaker s to an input vector $\overrightarrow{O_s} \in \mathfrak{R}^T$ and treat the training of CD-AFCPM as a mapping function Ψ_{AF}: $\mathfrak{R}^T \rightarrow \mathfrak{R}^{60G}$ (e.g. $\overrightarrow{A_s} = \Psi_{\mathrm{AF}}(\overrightarrow{O_s})$), we have an AF kernel of the form:

$$\widetilde{K}_{\mathrm{AF}}(\overrightarrow{O}_c, \overrightarrow{O}_s) = \left\langle \frac{\sqrt{\overrightarrow{w_b}} * \Psi_{\mathrm{AF}}(\overrightarrow{O_c})}{\sqrt{\Psi_{\mathrm{AF}}(\overrightarrow{O_b})}}, \frac{\sqrt{\overrightarrow{w_b}} * \Psi_{\mathrm{AF}}(\overrightarrow{O_s})}{\sqrt{\Psi_{\mathrm{AF}}(\overrightarrow{O_b})}} \right\rangle = \left\langle \Phi_{\mathrm{AF}}(\overrightarrow{O_c}), \Phi_{\mathrm{AF}}(\overrightarrow{O_s}) \right\rangle \tag{6.12}$$

where the mapping $\Phi_{\mathrm{AF}}(\cdot)$ is defined as:

$$\Phi_{\mathrm{AF}}(\overrightarrow{O}) = \frac{\sqrt{\overrightarrow{w_b}} * \Psi_{\mathrm{AF}}(\overrightarrow{O})}{\sqrt{\Psi_{\mathrm{AF}}(\overrightarrow{O_b})}}. \tag{6.13}$$

Figures 6.6(a) and 6.6(b) show the un-normalized $\overrightarrow{A_s}$ and the normalized $\overrightarrow{A_s}$ (i.e., $\varphi(\overrightarrow{A_s})$) for 150 speakers, respectively. For clarity, only feature elements with indexes between 531 and 660 are shown. Evidently, without normalization, some features have a large but almost constant value across all speakers (e.g., rows with dark-red color). These features will cause problems in SVM classification because they will dictate the decision boundary of the SVM, even though they contain little speaker-

dependent information. This problem has been largely alleviated by the normalization, as demonstrated in Figure 6.6(b). In particular, the normalization has the effect of keeping all features within a comparable range, which helps prevent the large but almost constant features from dominating the classification decision. Figures 6.7 shows the scoring procedure during the verification phase. Comparing Figure 6.7 and Figure 6.5 suggests that scoring based on the AF-kernel is more general. The scores produced by the AF-kernel may also complement the ones produced by the LR-based method, which will be demonstrated in the next section.

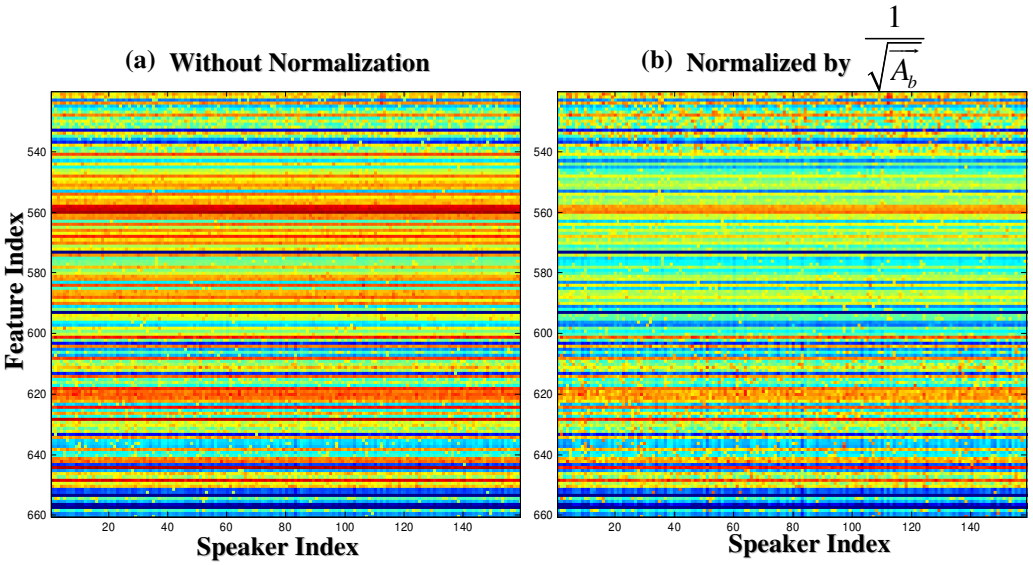
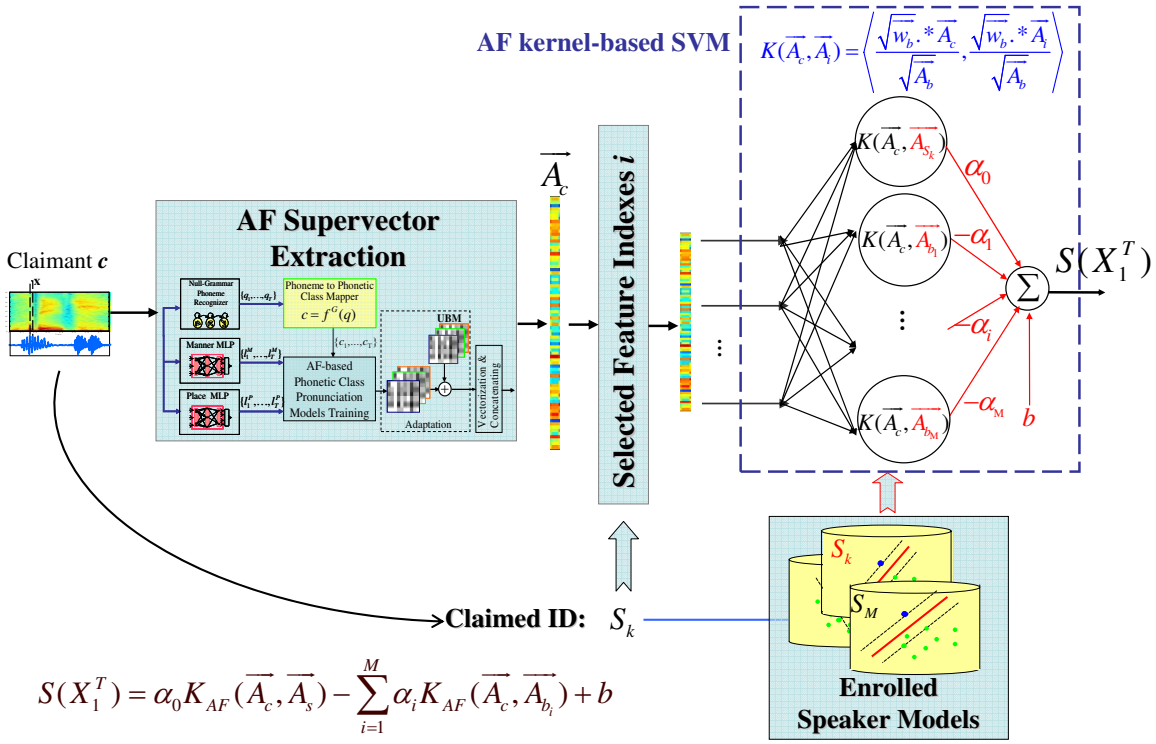


Figure 6.6: The effect of the normalization term $\frac{1}{\sqrt{A_b}}$ in the mapping $\varphi(\cdot)$.

6.4.3 Comparing AF-Kernel Scoring and LR-scoring

The SVM output of Fig. 6.7 can be considered as a scoring function:

$$S_{\text{AF-kernel}}(X_1^T) = \alpha_0 K_{\text{AF}}\left(\overrightarrow{A}_c, \overrightarrow{A}_s\right) - \sum_{i=1}^{M} \alpha_i K_{\text{AF}}\left(\overrightarrow{A}_c, \overrightarrow{A}_{b_i}\right) + b, \tag{6.14}$$



$$S(X_1^T) = \alpha_0 K_{AF}(\vec{A}_c, \vec{A}_s) - \sum_{i=1}^{M} \alpha_i K_{AF}(\vec{A}_c, \vec{A}_{b_i}) + b$$

Figure 6.7: The verification phase of an AF-kernel based speaker verification system.

where K_{AF} is the AF-kernel we derived, α_0 is the Lagrange multiplier corresponding to the target speaker, and α_i $(i = 1, \ldots, M)$ are Lagrange multipliers (some of them may be zero) corresponding to the background speakers. Comparing Eqs. 6.5 and 6.14 and comparing Figs. 6.5 and 6.7 suggest that AF-kernel scoring is more general and is potentially better than LR scoring (Eq. 6.5) in two aspects. First, the SVM optimally selects the most appropriate background speakers through the non-zero α_i. Second, instead of using a single background model that contains the average characteristics of all background speakers, a specific set of background speakers is used for each target speaker for scoring. This is to some extends analogous to cohort scoring. However, the cohort set is now discriminatively and optimally determined by SVM training, and the contribution of the selected background models is also optimally weighted through the Lagrange multipliers α_i.

6.5 Experiments and Results

6.5.1 Procedures

The training data are identical to those in Section 3.4.1. However, only the female part of NIST00 was used for evaluation. Moreover, unlike the training procedure in Section 3.4.2 and the scoring procedure in Eq. 3.12, each target speaker is represented by a supervector $\overrightarrow{A_s}$ and verification scoring is based on Eq. 6.10.

6.5.2 Score Fusion

Because AF-kernel scoring and LR-scoring are very different, they produce scores with with different dynamic range. Therefore, score normalization should be applied to achieve a similar dynamic range before fusion:

$$S_{\mathrm{F}}(X) \;=\; \alpha_u \frac{S_{\text{AF-kernel}}(X) - \mu_{\text{AF-kernel}}}{\sigma_{\text{AF-kernel}}} + (1-\alpha_u)\frac{S_{\mathrm{LR}}(X) - \mu_{\mathrm{LR}}}{\sigma_{\mathrm{LR}}} \tag{6.15}$$

where μ and σ are the mean and standard deviation of scores.

Figure 6.8 shows that normalizing the scores before fusion can make the EER less sensitive to the fusion weight α_u. Another advantage of score fusion is that the value of α_u can suggest which set of scores is more reliable. For example, in Figure 6.8, the scores produced by the GMM-UBM system are more reliable because the best fusion weight is about 0.4. However, Figure 6.8 also shows that fusion with or without normalization achieve almost the same EER.

6.5.3 Results and Discussions

The results of using AF-kernels (without feature selection) for computing the verification scores are shown in Table 6.1. Evidently, normalization helps reduce the EER significantly. Similar to the results in LR-based scoring approach, CD-AFCPM is superior to PD-AFCPM under the AF-kernel framework. Comparing the 7th row of Table 3.7 (EER = 23.46%) and the 2nd row of Table 6.1 (EER = 24.14%) suggests

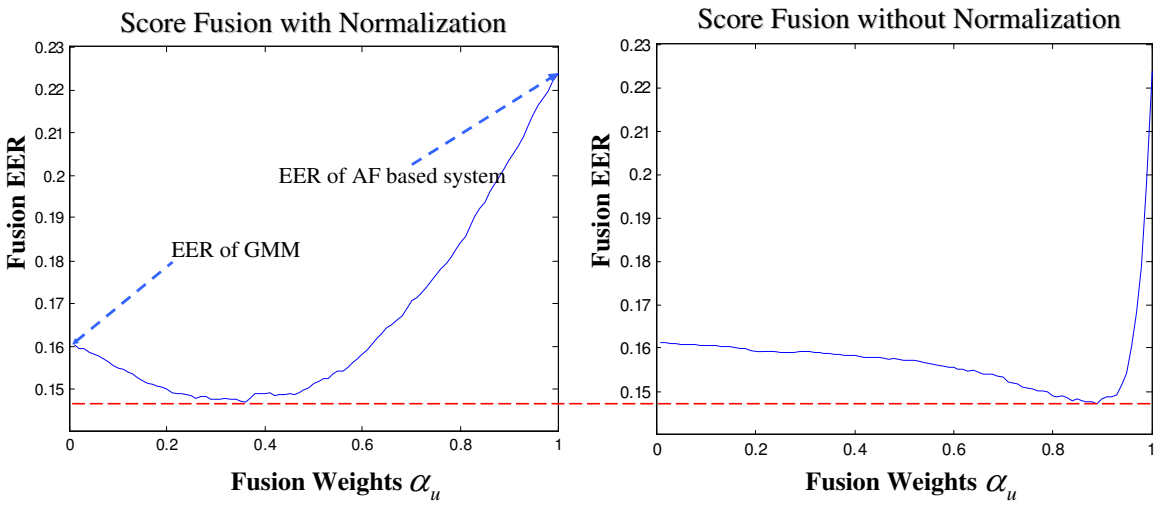


Figure 6.8: Score fusion with and without normalization.

that without feature selection, the AF-kernel is inferior to the conventional LR-based scoring.

Kernel Function	CD-AFCPM Supervector	PD-AFCPM Supervector
$K(\vec{A}_c, \vec{A}_s) = \langle \vec{A}_c, \vec{A}_s \rangle$	26.12%	28.63%
$K(\vec{A}_c, \vec{A}_s) = \left\langle \frac{\sqrt{\vec{w}_b} .* \vec{A}_c}{\sqrt{\vec{A}_b}}, \frac{\sqrt{\vec{w}_b} .* \vec{A}_s}{\sqrt{\vec{A}_b}} \right\rangle$	24.14%	27.14%

Table 6.1: The EERs of AF kernel-based speaker verification systems using PD-AFCPM and CD-AFCPM supervectors without feature selection.

Kernel Function	No Feature Selection	Feature Selection
$K(\vec{A}_c, \vec{A}_s) = \left\langle \frac{\sqrt{\vec{w}_b} .* \vec{A}_c}{\sqrt{\vec{A}_b}}, \frac{\sqrt{\vec{w}_b} .* \vec{A}_s}{\sqrt{\vec{A}_b}} \right\rangle$	24.14%	23.87%

Table 6.2: EER achieved by the AF kernel-based speaker verification system using CD-AFCPM supervectors with and without feature selection.

We conjecture that the inferiority is caused by the irrelevant features in the supervectors. To verify this conjecture, we performed the same experiment but with the irrelevant features removed by SVM-RFE [63]. The results are shown in Table 6.2. Evidently, selecting relevant features can improve the performance.

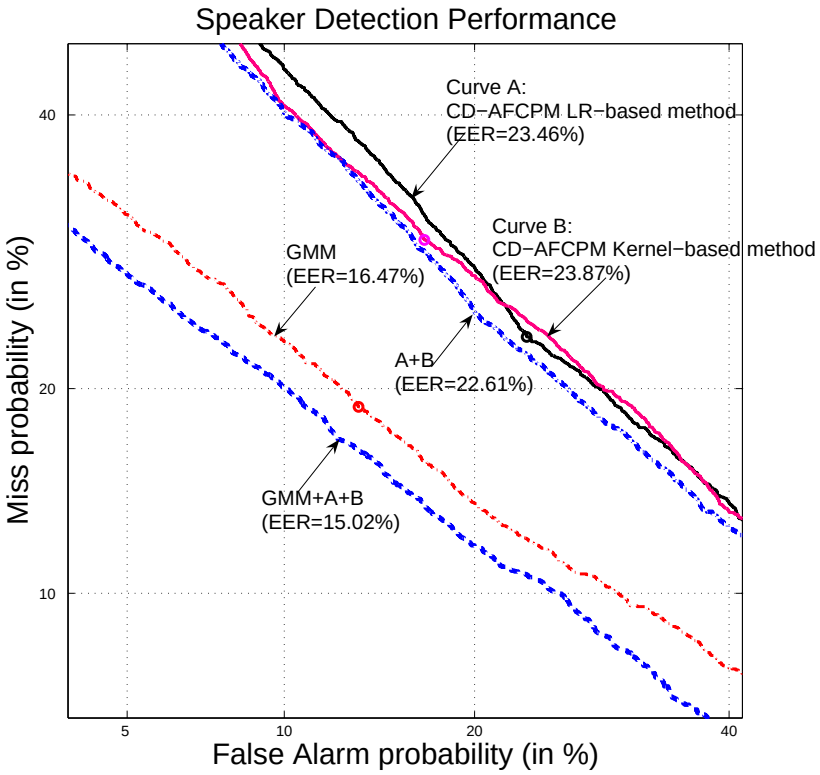


Figure 6.9: DET produced by LR scoring, AF-kernel scoring, acoustic GMM-UBM, and their fusions.

Figure 6.9 shows the DET curves of different scoring methods and their fusion with a GMM-UBM system. The results show that scoring based on the AF-kernel K_{AF} (Curve B) outperforms LR scoring (Curve A) at the low false-alarm region, whereas the situation is reverse at the low miss-probability region. This suggests that the two scoring methods are complementary to each other, which is evident by the superior performance (Curve A+B) when the scores resulting from the two scoring methods

are fused. The p-values [48] between the EERs of the fusion and non-fusion cases are all smaller than 0.00001, suggesting that the differences in EERs are statistically significant.

At the low-miss probability region, AF-kernel scoring is only slightly worse than LR scoring, but it is significantly better than LR scoring in the low false alarm region. This suggests that AF-kernel scoring is generally better than LR scoring, which is mainly attributed to the explicitly use of discriminative information in the kernel function of the SVM and to the optimal selection of background speakers by SVM training. Although LR scoring also considers the impostor information, it can only implicity use this information through the UBM. In AF-kernel scoring, on the other hand, the SVM of each target speaker is discriminatively trained to differentiate the target speaker from all of the background speakers. The SVM effectively provides an optimal set of weights for this differentiation. On the other hand, in log-likelihood scoring, all target speakers share the same background model and the weight is always equal ($= -1$) across all target speakers. This explains the superiority of the AF-kernel scoring approach.

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This dissertation addresses three important issues in text-independent speaker verification: data-spareness, robustness, and discriminative power of speaker models. To these ends, this dissertation proposes four new techniques for articulatory-feature based pronunciation modeling, including phonetic-class dependent AFCPM (CD-AFCPM), probabilistic-weighted CD-AFCPM (PW-CD-AFCPM), model adaptation and articulatory-feature kernels. These four techniques have been evaluated on the NIST 2000 dataset. The results show that: (1) combining the classical phoneme tree and Euclidean distance between AFCPMs is a promising way to map phonemes to phonetic classes in CD-AFCPM; (2) phonetic-classes AFCPM achieves a significantly lower error rate as compared to conventional AFCPM; (3) performance can be slightly improved by weighting the CD-AFCPM according to the probability of observing the phonetic classes; (4) better AFCPMs can be created by not only adapting the background models but also the phoneme-independent speaker models; and (5) AF-kernel scoring is complementary to likelihood-ratio scoring, and their fusion can improve verification performance.

7.2 Future Work

7.2.1 Robustness Analysis of Articulatory Pronunciation Modeling for High-Level Speaker Verification

One possible extension of this work is to analyze the performance of AFCPMs when the accuracy of the articulatory features extractor varies. Figure 7.1 shows the EERs achieved by the CD-AFCPMs and PD-AFCPMs for three different the accuracies of the manner and place MLPs. Figure 7.1 shows that the best performing MLP does not lead to the lowest EER. This suggests that having high accuracy in the MLPs does not necessarily mean high speaker recognition performance. We suspect that this is mainly because the main purpose of the MLPs is to capture the articulatory features of speakers instead of classifying the articulatory properties. Therefore, as long as the patterns of mistakes made by these MLPs are consistent for the same speaker and different for different speakers, they can still provide valuable speaker information for building the pronunciation models. This conjecture is supported by the experimental results shown in the Figure 7.1.

We can see from Figure 7.1 that both PD-AFCPM and CD-AFCPM are robust to the articulatory features extractors. Currently, the MLPs use 9 consecutive MFCC frames as a unit. It is also of interest to see how the window size affects the accuracy of these MLPs, which in turn may affect the modeling capability of AFCPMs.

The phoneme recogizer is another important component of the pronunciation modeling investigated in this work. We conjecture that the recognizer's accuracy will have effect on the performance of AFCPMs. Another interesting extension of this work is to replace the null-grammar recognizer with a full-blown speech recognizer equipped with a good language model. Because a good language model will help the recognizer to "correct" the pronunciation mistakes made by a speaker, the performance of AFCPMs may degrade if the langauge model is too perfect. Therefore, it is interesting to find the best compromise between having no language model (i.e. null-grammar)

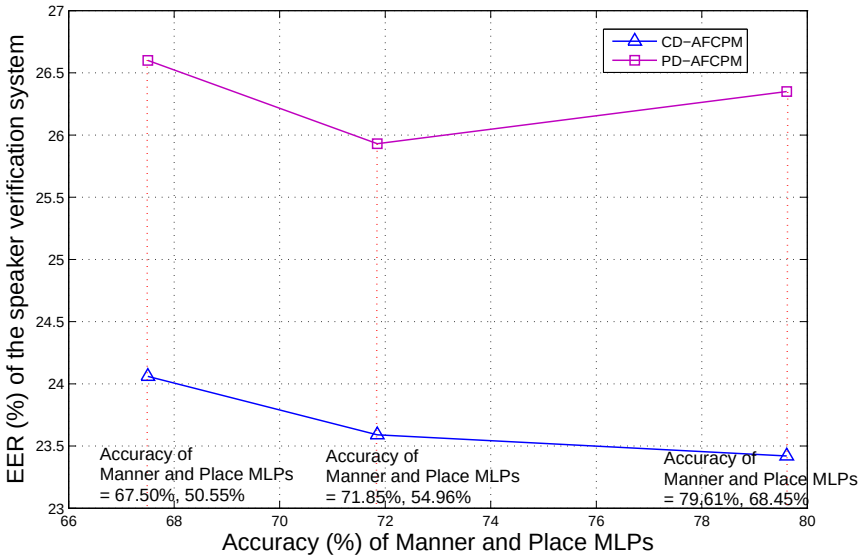


Figure 7.1: EERs achieved by the CD-AFCPM and PD-AFCPM at different accuracies of the manner and place MLPs.

and having an almost perfect language model in the phoneme recognizer. To achieve this, we can systematically degrade the accuracy of a good language model and see how the degradation affects the performance of the AFCPMs.

7.2.2 Derive the AF-Kernels from Similarity Score

One way to derive kernels is to use similarity scores. In order to compute the similarity scores a discriminant function must be constructed. Note that the AF kernel in Section 6.4 is based on the discriminant function (scoring function):

$$\begin{aligned} S_{\text{CD-AFCPM}}(X_1^T) &= \frac{1}{T}\sum_{t=1}^{T}\left(\log \widehat{p}_s^{\text{CD}}(X_t) - p_b^{\text{CD}}(X_t)\right) \\ &= \left\langle \log \frac{\overrightarrow{A_s}}{\overrightarrow{A_b}}, \overrightarrow{A_c'} \right\rangle . \end{aligned}$$

However, the kernel can be derived from a more general discriminant function $S(X_1^T) = f_s(\overrightarrow{A_c})$, where $f_s(\overrightarrow{A_c})$ can be any functions used for computing the score between the utterances of target speaker s and claimant c. Our goal is therefore to find the discriminant function $f_s(\overrightarrow{A_c})$.

Assume that there are two sets of training data $\left\{\overrightarrow{A_s}, y_s = +1\right\}$ and $\left\{\overrightarrow{A_{b_k}}, y_{b_k} = -1\right\}_{k=1}^{M}$. Then, the problem of finding a discriminant function can be formulated as:

$$\min_{f_s \in \mathcal{R}_K} \left\{ \sum_{i \in \{s, b_k\}_{k=1}^{M}} L\left(f_s(\overrightarrow{A_i}), y_i\right) + \lambda \parallel f_s \parallel^2 \right\} \tag{7.1}$$

where $\mathcal{R}_K$ is the Reproducing Kernel Hilbert Space (RKHS) [64], λ is a penalty factor, and $L\left(f_s(\overrightarrow{A_i}), y_i\right) = \rho_i(y_i - f_s(\overrightarrow{A_i}))^2$ is a loss function, where ρ_i is used to solve the data unbalance problem. The reason for searching f_s in $\mathcal{R}_K$ is that only in RKHS the solution of the optimization problem (Eq. 7.1) can be parameterized as a linear combination of the training data [64]:

$$f_s(\overrightarrow{A_c}) = \sum_{i=1}^{M+1} w_i k(\overrightarrow{A_c}, \overrightarrow{A_i}), \tag{7.2}$$

where $k(\cdot, \cdot)$ is a positive definite kernel belonging to the function space $\mathcal{R}_K$ such that

$$\langle f_s(\cdot), k(\overrightarrow{x}, \cdot) \rangle_{\mathcal{R}_K} = f_s(\overrightarrow{x}). \tag{7.3}$$

Based on Eq. 7.2 and Eq. 7.3, we have:

$$\begin{aligned} \| f_s \|^2 = \langle f_s, f_s \rangle &= \left\langle f_s, \sum_{i=1}^{M+1} w_i k(\overrightarrow{A_i}, \cdot) \right\rangle \\ &= \sum_{i=1}^{M+1} w_i \left\langle f_s(\cdot), k(\overrightarrow{A_i}, \cdot) \right\rangle = \sum_{i=1}^{M+1} w_i f_s(\overrightarrow{A_i}) \\ &= \sum_{i=1}^{M+1} w_i \left(\sum_{j=1}^{M+1} w_j (k(\overrightarrow{A_i}, \overrightarrow{A_j})) \right). \end{aligned} \tag{7.4}$$

Therefore, the optimization problem in Eq. 7.1 can be formulated as:

$$\min_{\overrightarrow{w}} \left\{ (\overrightarrow{Y} - \mathbf{K}\overrightarrow{w})^{\mathrm{T}} \mathbf{\Lambda} (\overrightarrow{Y} - \mathbf{K}\overrightarrow{w}) + \lambda \overrightarrow{w}^{\mathrm{T}} \mathbf{K} \overrightarrow{w} \right\} \tag{7.5}$$

where

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,M+1} \\ k_{2,1} & \cdots & \cdots & k_{2,M+1} \\ \vdots & \vdots & k_{i,j} & \vdots \\ k_{M+1,1} & k_{M+1,2} & \cdots & k_{M+1,M+1} \end{bmatrix}$$

with $k_{i,j} = k(\overrightarrow{A_i}, \overrightarrow{A_j})$,

$$\mathbf{\Lambda} = \begin{bmatrix} \rho_1 & 0 & \cdots & 0 \\ 0 & \rho_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \rho_{M+1} \end{bmatrix}$$

and

$$\overrightarrow{Y} = \begin{bmatrix} 1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}_{(M+1)\times 1} .$$

The optimization problem in Eq. 7.5 can be solved by

$$\begin{aligned} &\frac{\partial}{\partial \overrightarrow{w}} \left\{ (\overrightarrow{Y} - \mathbf{K}\overrightarrow{w})^{\mathrm{T}} \mathbf{\Lambda} (\overrightarrow{Y} - \mathbf{K}\overrightarrow{w}) + \lambda \overrightarrow{w}^{\mathrm{T}} \mathbf{K} \overrightarrow{w} \right\} = 0 \\ &\Longrightarrow \overrightarrow{w} = (\mathbf{K}^{\mathrm{T}} \mathbf{\Lambda} \mathbf{K} + \lambda \mathbf{K}^{\mathrm{T}})^{-\mathbf{1}} (\mathbf{K}^{\mathrm{T}} \mathbf{\Lambda} \overrightarrow{Y}). \end{aligned} \tag{7.6}$$

As a result, we can derive the discriminant function and compute the similarity score as:

$$\begin{aligned} &Similarity(\overrightarrow{A_c}, \overrightarrow{A_s}) = f_s(\overrightarrow{A_c}) = \sum_{i=1}^{M+1} w_i k(\overrightarrow{A_c}, \overrightarrow{A_i}) \\ &= \left[(\mathbf{K}^{\mathrm{T}} \mathbf{\Lambda} \mathbf{K} + \lambda \mathbf{K}^{\mathrm{T}})^{-\mathbf{1}} (\mathbf{K}^{\mathrm{T}} \mathbf{\Lambda} \overrightarrow{Y}) \right]^{\mathrm{T}}_{(M+1)\times 1} \begin{bmatrix} k(\overrightarrow{A_c}, \overrightarrow{A_s}) \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_M}}) \end{bmatrix} . \end{aligned} \tag{7.7}$$

By dropping the constant in the similarity score, Eq. 7.7 can be approximated as:

$$
\begin{aligned}
Similarity(\overrightarrow{A_c}, \overrightarrow{A_s}) &\approx \begin{bmatrix} k(\overrightarrow{A_s}, \overrightarrow{A_s}) \\ k(\overrightarrow{A_s}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_s}, \overrightarrow{A_{b_M}}) \end{bmatrix}^{\mathrm{T}} (\mathbf{K}\mathbf{\Lambda}\mathbf{K}^{\mathrm{T}} + \lambda\mathbf{K})^{-1} \begin{bmatrix} k(\overrightarrow{A_c}, \overrightarrow{A_s}) \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_M}}) \end{bmatrix} \\
&= \left\langle (\mathbf{K}\mathbf{\Lambda}\mathbf{K}^{\mathrm{T}} + \lambda\mathbf{K})^{-\frac{1}{2}} \begin{bmatrix} k(\overrightarrow{A_s}, \overrightarrow{A_s}) \\ k(\overrightarrow{A_s}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_s}, \overrightarrow{A_{b_M}}) \end{bmatrix}, (\mathbf{K}\mathbf{\Lambda}\mathbf{K}^{\mathrm{T}} + \lambda\mathbf{K})^{-\frac{1}{2}} \begin{bmatrix} k(\overrightarrow{A_c}, \overrightarrow{A_s}) \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_M}}) \end{bmatrix} \right\rangle.
\end{aligned}
$$

7.2.3 Derive the AF-Kernel from Distance Metric

In addition to deriving the kernel function based on similarity measures, the AF-kernel can also be derived from a distance metric. This idea is inspired from an alternative expression of Eq. 6.5:

$$
\begin{aligned}
S_{\text{CD-AFCPM}}(X_1^T) &= \frac{1}{T}\sum_{t=1}^{T}\left(\log\frac{\widehat{p}_s^{\,\mathrm{CD}}(X_t)}{p_b^{\mathrm{CD}}(X_t)}\right) \\
&= \left\langle \overrightarrow{A_c'}, \log\frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \right\rangle = \left\langle \overrightarrow{A_c'}, \log\frac{\overrightarrow{A_s}\overrightarrow{A_c'}}{\overrightarrow{A_b}\overrightarrow{A_c'}} \right\rangle \\
&= \left\langle \overrightarrow{A_c'}, \log\frac{\overrightarrow{A_c'}}{\overrightarrow{A_b}} \right\rangle - \left\langle \overrightarrow{A_c'}, \log\frac{\overrightarrow{A_c'}}{\overrightarrow{A_s}} \right\rangle \\
&= G\cdot\mathcal{D}_{\mathrm{KL}}(\frac{\overrightarrow{A_c'}}{G} \parallel \frac{\overrightarrow{A_b}}{G}) - G\cdot\mathcal{D}_{\mathrm{KL}}(\frac{\overrightarrow{A_c'}}{G} \parallel \frac{\overrightarrow{A_s}}{G}) \\
&= \mathcal{D}_{\mathrm{KL}}(\overrightarrow{A_c'} \parallel \overrightarrow{A_b}) - \mathcal{D}_{\mathrm{KL}}(\overrightarrow{A_c'} \parallel \overrightarrow{A_s})
\end{aligned}
\tag{7.8}
$$

where $\mathcal{D}_{\mathrm{KL}}(\overrightarrow{p_1} \parallel \overrightarrow{p_2})$ is the Kullback-Leibler (KL) divergence of the density functions $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$.[1] The implementation of Eq. 7.8 is shown in Figure 7.2. Note that instead of computing the dot products inside the processing nodes (circles) of the LR-scoring block in Figure 6.5, $\mathcal{D}_{\mathrm{KL}}(\overrightarrow{A_c'} \parallel \overrightarrow{A_s})$ is computed and the weights '+1' and '−1' are reversed.

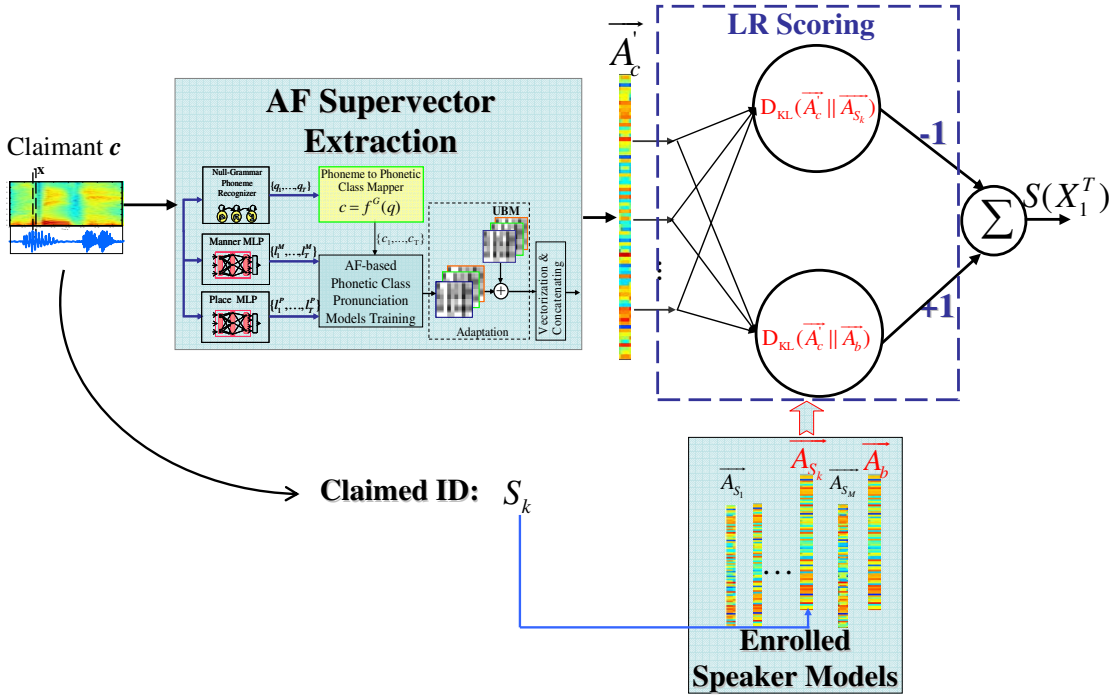


Figure 7.2: Implementing the traditional log-likelihood scoring of CD-AFCPM speaker verification in KL divergence form.

Comparing Eq. 6.5 (Figure 6.5) and Eq. 7.8 (Figure 7.2) suggests that the kernel can also be derived from a distance metric. To this end, we define $\mathcal{D}(\mathrm{utt}_c \parallel \mathrm{utt}_s)$ as the distance between two utterances utt_c and utt_s. Then, we assume that

$$\mathcal{D}(\mathrm{utt}_c \parallel \mathrm{utt}_s) \approx \sqrt{K(\overrightarrow{O_c}, \overrightarrow{O_c}) - 2K(\overrightarrow{O_c}, \overrightarrow{O_s}) + K(\overrightarrow{O_s}, \overrightarrow{O_s})}, \tag{7.9}$$

[1] Although strictly speaking $\overrightarrow{A_c'}$ cannot be treated as a density functions ($\frac{\overrightarrow{A_c'}}{G}$ can), the Eq. 7.8 is numerically correct. We use the symbol here for readability and consistency.

where $K(\cdot,\cdot)$ is a kernel function. Note that the right part of Eq. 7.9 is actually the distance between the mapping vectors $\Psi(\overrightarrow{O_c})$ and $\Psi(\overrightarrow{O_s})$ in the kernel-induced feature space [64]. Therefore, if we can compute the distance between utterances c and s, we can derive the kernel function from Eq. 7.9.

For example, if the distance between utterances c and s is given by:

$$\mathcal{D}^2(\mathrm{utt}_c \parallel \mathrm{utt}_s) = \|\overrightarrow{A_c} - \overrightarrow{A_s}\|^2 = \overrightarrow{A_c}^{\mathrm{T}}\overrightarrow{A_c} - 2\overrightarrow{A_c}^{\mathrm{T}}\overrightarrow{A_s} + \overrightarrow{A_s}^{\mathrm{T}}\overrightarrow{A_s}, \tag{7.10}$$

then, using Eq. 7.9, we have

$$\overrightarrow{A_c}^{\mathrm{T}}\overrightarrow{A_c} - 2\overrightarrow{A_c}^{\mathrm{T}}\overrightarrow{A_s} + \overrightarrow{A_s}^{\mathrm{T}}\overrightarrow{A_s} = K(\overrightarrow{O_c},\overrightarrow{O_c}) - 2K(\overrightarrow{O_c},\overrightarrow{O_s}) + K(\overrightarrow{O_s},\overrightarrow{O_s}). \tag{7.11}$$

As a result, the solution (function) of Eq. 7.10 can be expressed as an AF kernel:

$$K_{\mathrm{AF}}(\overrightarrow{O_c},\overrightarrow{O_s}) = \left\langle \Psi_{\mathrm{AF}}(\overrightarrow{O_c}), \Psi_{\mathrm{AF}}(\overrightarrow{O_s}) \right\rangle = \left\langle \overrightarrow{A_c}, \overrightarrow{A_s} \right\rangle. \tag{7.12}$$

We can see that if the distance between utterances c and s is computed as the Euclidean distance between the two corresponding AF supervectors, the AF kernel becomes a linear kernel in the space spanned by the AF supervectors.

Many other distance metrics can be applied as well. For example, if the Mahalanobis distance of AF supervectors is used as the distance metric between two utterances, we obtain [58]:

$$K_{\mathrm{AF}}(\overrightarrow{O_c},\overrightarrow{O_s}) = \left\langle \Sigma^{-\frac{1}{2}}\overrightarrow{A_c}, \Sigma^{-\frac{1}{2}}\overrightarrow{A_s} \right\rangle = \left\langle \Sigma^{-\frac{1}{2}}\Psi_{\mathrm{AF}}(\overrightarrow{O_c}), \Sigma^{-\frac{1}{2}}\Psi_{\mathrm{AF}}(\overrightarrow{O_s}) \right\rangle, \tag{7.13}$$

where Σ is the covariance matrix of the input supervectors. More generally, we can even train a scaling and rotation matrix Q to improve the robustness of the system

against the mismatch and distortion in the utterances, which leads to:

$$\mathcal{D}^2(\mathrm{utt}_c \parallel \mathrm{utt}_s) = (\overrightarrow{A_c} - \overrightarrow{A_s})^{\mathrm{T}} Q^{-1} (\overrightarrow{A_c} - \overrightarrow{A_s}). \tag{7.14}$$

Therefore the AF kernel becomes:

$$K_{\mathrm{AF}}(\overrightarrow{O_c}, \overrightarrow{O_s}) = \left\langle Q^{-\frac{1}{2}} \overrightarrow{A_c}, Q^{-\frac{1}{2}} \overrightarrow{A_s} \right\rangle = \left\langle Q^{-\frac{1}{2}} \Psi_{\mathrm{AF}}(\overrightarrow{O_c}), Q^{-\frac{1}{2}} \Psi_{\mathrm{AF}}(\overrightarrow{O_s}) \right\rangle. \tag{7.15}$$

BIBLIOGRAPHY

[1] Elizabeth Shriberg, "Higher-level features in speaker recognition," *Speaker Classification*, vol. 1, pp. 241–259, 2007.

[2] J. R. Deller Jr, J. G. Proakis, and J. H. L. Hansen, *Discrete-time Processing of Speech Signals*, Macmillan Pub. Company, 1993.

[3] S. Y. Kung, M. W. Mak, and S. H. Lin, *Biometric Authentication: A Machine Learning Approach*, Prentice Hall, Upper Saddle River, New Jersey, 2005.

[4] J. P. Campbell Jr., "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[5] S. Furui, "Recent advances in speaker recognition," *Pattern Recognition Letters*, vol. 18, pp. 859–872, 1997.

[6] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[7] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, Apr 1984.

[8] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech'97*, 1997, pp. 1895–1898.

[9] *http://www.nist.gov/speech/tests/spk/*" .

[10] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *Proc. ICASSP*, 2003, vol. 2, pp. 6–10.

[11] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[12] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, pp. 42–54, 2000.

[13] A. Schmidt-Nielsen and T. H. Crystal, "Speaker verification by human listeners: Experiments comparing human and machine performance using the NIST 1998 speaker evaluation data," *Digital Signal Processing*, vol. 10, pp. 249–266, 2000.

[14] E. Blaauw, "The contribution of prosodic boundary markers to the perceptual difference between read and spontaneous speech," *Speech Communication*, vol. 14, pp. 359–375, 1994.

[15] D. Dahan and J. M. Bernard, "Interspeaker variability in emphatic accent production in French," *Language and Speech*, vol. 39, no. 4, pp. 341–374, 1996.

[16] J. Sussman, E. Dalston, and S. Gumbert, "The effect of speaking style on a locus equation characterization of stop place articulation," *Phonetica*, vol. 55, no. 4, pp. 204–255, 1998.

[17] D. P. Kuehn and K.L. Moll, "A cineradiographic study of VC and CV articulatory velocities," *J. Phonetics*, vol. 23, no. 4, pp. 303–320, 1976.

[18] E. Shriberg, et al., "Modeling prosodic sequences for speaker recognition," *Speech Communication*, vol. 4, pp. 455–472, 2005.

[19] B. Peskin, J. Navratil, J. Abramson, D. Jones, D. Klusáček, D. Reynolds, and B. Xiang, "Using prosodic and conversational features for high-performance speaker recognition: Report from JHU WS'02," in *Proc. ICASSP*, 2003, vol. 4, pp. 792–795.

[20] A. Adami, R. Mihaescu, D. Reynolds, and J. Godfrey, "Modeling prosodic dynamics for speaker recognition," in *Proc. ICASSP*, 2003, vol. 4, pp. 788–791.

[21] K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub, "Modeling dynamic prosodic variation for speaker verification," in *ICSLP*, 1998, vol. 4, pp. 3189–3192.

[22] F. Weber, L. Manganaro, B. Peskin, and E. Shriberg, "Using prosodic and lexical information for speaker identification," in *Proc. ICASSP*, 2002, vol. 1, pp. 141–144.

[23] M. J. Carey, E. S. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust prosodic features for speaker identification," in *ICSLP*, 1996, vol. 3, pp. 1800–1803.

[24] K. Bartkova, D. Le-Gac, D. Charlet, and D. Jouvet, "Prosodic parameter for speaker identification," in *ICSLP*, 2002, vol. 1, pp. 1197–1200.

[25] D. Reynolds, et. al., "The superSID project: Exploiting high-level information for high-accuracy speaker recognition," in *Proc. ICASSP*, Hong Kong, April 2003, vol. 4, pp. 784–787.

[26] B. S. Atal, "Automatic speaker recognition based on pitch contours," *J. Acoust. Soc. Am.*, vol. 52, pp. 1687–1972, 1972.

[27] D. Chappell and J. Hansen, "Speaker-specific pitch contour modeling and modification," in *Proc. ICASSP*, 1998, vol. 1, pp. 885–888.

[28] G. R. Doddington, "Speaker recognition based on idiolectal differences between speakers," in *Proc. Eurospeech*, Aalborg, Sept. 2001, pp. 2521–2524.

[29] S. Kajarekar, L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman, and J. Zheng, "Speech recognition performance comparison between DSR and AMR transcoded speech," in *Proc. ICASSP*, 2005, vol. 1, pp. 173– 176.

[30] W. Andrews, et al., "Gender-dependent phonetic refraction for speaker recognition," in *Proc. ICASSP*, 2002.

[31] J. P. Campbell, D. A. Reynolds, and R. B. Dunn, "Fusing high- and low-level features for speaker recognition," in *Proc. Eurospeech*, 2003, pp. 2665–2668.

[32] J. Navratil, Q. Jin, W. Andrews, and J. Campbell, "Phonetic speaker recognition using maximum likelihood binary decision tree models," in *Proc. ICASSP*, 2003, vol. 4, pp. 796–799.

[33] Q. Jin, et al., "Combining cross-stream and time dimensions in phonetic speaker recognition," in *Proc. ICASSP*, 2003.

[34] D. Klusacek, J. Navratil, D. A. Reynolds, and J. P. Campbell, "Conditional pronunciation modeling in speaker detection," in *Proc. ICASSP*, 2003, vol. 4, pp. 804–807.

[35] K. Y. Leung, M. W. Mak, M. H. Siu, and S. Y. Kung, "Adaptive articulatory feature-based conditional pronunciation modeling for speaker verification," *Speech Communication*, vol. 48, no. 1, pp. 71–84, 2006.

[36] K. Kirchhoff, *Robust Speech Recognition Using Articulatory Information*, PhD thesis, University of Bielefeld, 1999.

[37] P. Frber, "Quicknet on multispert: Fast parallel neural network training," Tech. Rep. TR-97-047, ICSI, 1998.

[38] R. Auckenthaler, E. Parris, and M. Carey, "Improving a GMM speaker verification system by phonetic weighting," in *Proc. ICASSP*, 1999, pp. 1440–1444.

[39] K. K. Yiu, M. W. Mak, M. C. Cheung, and S. Y. Kung, "Blind stochastic feature transformation for channel robust speaker verification," *J. of VLSI Signal Processing*, vol. 42, no. 2, pp. 117–126, 2006.

[40] "The NIST year 1999 speaker recognition evaluation plan," in *http://www.nist.gov/speech/tests/spk/1999/doc.*

[41] "The NIST year 2000 speaker recognition evaluation plan," in *http://www.nist.gov/speech/tests/spk/2000/doc.*

[42] J. P. Campbell and D. A. Reynolds, "Corpora for the evaluation of speaker recognition systems," in *Proc. ICASSP*, 1999, vol. 2, pp. 829–832.

[43] D. A. Reynolds, "HTIMIT and LLHDB: Speech corpora for the study of handset transducer effects," in *Proc. ICASSP*, 1997, vol. 2, pp. 1535–1538.

[44] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on ASSP*, vol. 28, no. 4, pp. 357–366, August 1980.

[45] M. W. Mak, K. K. Yiu, and S. Y. Kung, "Probabilistic feature-based transformation for speaker verification over telephone networks," *Neurocomputing, special issue on Neural Networks for Speech and Audio Processing*, 2007.

[46] S. Y. Kung and M. W. Mak, "Machine learning for multi-modality genomic signal processing," *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 117–121, May 2006.

[47] S. H. Lin, S. Y. Kung, and L. J. Lin, "A probabilistic DBNN with applications to sensor fusion and object recognition," in *Proc. 5th IEEE Workshop on Neural Networks for Signal Processing*, Cambridge, MA, Aug. 1995, pp. 333–342.

[48] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, 1989, pp. 532–535.

[49] B. Efron, "Estimating the error rate of a prediction rule: Improvement on cross-validation,," *American Statistical Association*, vol. 78, pp. 316–331, 1983.

[50] B. Mak, S. Ho, R. Hsiao, and J. T. Kwok, "Embedded kernel eigenvoice speaker adaptation and its implication to reference speaker weighting," *IEEE Transactions on Speech and Audio Processing*, vol. 14, pp. 1267– 1280, 2006.

[51] T. Matsui and S. Furui, "Concatenated phoneme models for text-variable speaker recognition," in *Proc. ICASSP*, 1993, vol. 1, pp. 391–394.

[52] M. W. Mak, R. Hsiao, and B. Mak, "A comparison of various adaptation methods for speaker verification with limited enrollment data," in *ICASSP*, 2006, pp. 929–932.

[53] K. Y. Leung, M. W. Mak, and S. Y. Kung, "Adaptive articulatory feature-based conditional pronunciation modeling for speaker verification," *Speech Communication*, vol. 48, no. 1, pp. 71–84, 2006.

[54] S. X. Zhang, M. W. Mak, and Helen H. Meng, "Speaker verification via high-level feature based phonetic-class pronunciation modeling," *IEEE Trans. on Computers*, vol. 56, no. 9, pp. 1189–1198, 2007.

[55] Jonathon Shlens, "A tutorial on principal component analysis," *http://www.snl.salk.edu/ shlens/pub/notes/pca.pdf*, 2005, December.

[56] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[57] W.M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc. ICASSP*, 2002, vol. 1, pp. 161–164.

[58] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308–311, 2006, May.

[59] V. Wan and S. Renals, "SVMSVM: Support vector machine speaker verification methodology," in *Proc. ICASSP*, 2003, vol. II, pp. 221–224.

[60] C. S. Liu, C. H. Lee, B. H. Juang, and A. E. Rosenberg, "Speaker recognition based on minimum error discriminative training," in *Proc. ICASSP*, 1994, vol. 1, pp. 325–328.

[61] A. E. Rosenberg, O. Siohan, and S. Parthasarathy, "Speaker verification using minimum verification error training," in *Proc. ICASSP*, 1998, pp. 105–108.

[62] Chengyuan Ma and Eric Chang, "Comparison of discriminative training methods for speaker verification," in *Proc. ICASSP*, 2003, April, vol. 1, pp. 192–195.

[63] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389–422, 2002.

[64] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambrige, 2004.

Appendix A

PHONEMES AND PHONETIC CLASSES

Tables A.1, A.2 and A.3 show the relationship between phoneme and phonetic classes obtained from the classical phoneme tree.

Phonetic Class c	**Phone Type**	**Phoneme** q
1	Vowels	iy, ih, ey, eh, ae, er, ax, ah, uw, uh, ow ao, aa
2	Fricatives	v, dh, z, zh, f, th, s, sh
3	Whisper	hh
4	Affricates	jh, ch
5	Diphthongs	ay, aw, oy
6	Semivowels	r, l, el, w, y
7	Consonants	b, d, g, p, t, k
8	Nasals	m, en, n, ng

Table A.1: The relationship between phonemes and phonetic classes obtained from the classical phoneme tree [2] when the total number of phonetic classes $G = 8$.

Phonetic Class c	**Phone Type**	**Phoneme** q
1	Front Vowels	iy, ih, ey, eh, ae
2	Mid Vowels	er, ax, ah
3	Back Vowels	uw, uh, ow, ao, aa
4	Voice Fricatives	v, dh, z, zh
5	Unvoiced Fricatives	f, th, s, sh
6	Whisper	hh
7	Affricates	jh, ch
8	Diphthongs	ay, aw, oy
9	Semivowels	r, l, el, w, y
10	Consonants	b, d, g, p, t, k
11	Nasals	m, en, n, ng

Table A.2: The relationship between phonemes and phonetic classes obtained from the classical phoneme tree [2] when $G = 11$.

Phonetic Class c	**Phone Type**	**Phoneme** q
1	Front Vowels	iy, ih, ey, eh, ae
2	Mid Vowels	er, ax, ah
3	Back Vowels	uw, uh, ow, ao, aa
4	Voiced Fricatives	v, dh, z, zh
5	Unvoiced Fricatives	f, th, s, sh
6	Whisper	hh
7	Affricates	jh, ch
8	Diphthongs	ay, aw, oy
9	Liquids Semivowels	r, l, el
10	Glides Semivowels	w, y
11	Voiced Consonants	b, d, g
12	Unvoiced Consonants	p, t, k
13	Nasals	m, en, n, ng

Table A.3: The relationship between phonemes and phonetic classes obtained from the classical phoneme tree [2] when $G = 13$.

Appendix B

PROOFS OF EQUATIONS

Proof of

$$\boxed{P_b(m,p|*) = \sum_{i=1}^{46} [P_b(m,p|q_i)P_b(q_i)]}$$

Assume:

$$\#\left((m,p,q_i)\text{in the utterance of all backgroud speakers}\right) = a_i$$

$$\#\left((*,*,q_i)\text{in the utterance of all backgroud speakers}\right) = b_i$$

where q_i represents one of 46 phoneme in English.

$\therefore$

$$P_b(m,p|q_i) = \frac{\#((m,p,q_i)\text{ in the utterances of all backgroud speakers }\text{)}}{\#((*,*,q_i)\text{ in the utterances of all backgroud speakers)}} = \frac{a_i}{b_i}$$

$$P_b(m,p|*) = \frac{\#((m,p,*)\text{ in the utterances of all backgroud speakers}}{\#((*,*,*)\text{ in the utterances of all backgroud speakers}} = \frac{\sum_{i=1}^{46} a_i}{\sum_{i=1}^{46} b_i} \tag{B.1}$$

Assume that there exists a constant A_i that satisfies

$$\sum_{i=1}^{46} [A_i P_b(m,p|q_i)] = P_b(m,p|*) \tag{B.2}$$

$\therefore$

$$\sum_{i=1}^{46}\left(A_i\frac{a_i}{b_i}\right)=\frac{\sum_{i=1}^{46}a_i}{\sum_{i=1}^{46}b_i}$$

$$\sum_{i=1}^{46}\left(A_i\frac{a_i\left(\sum_{i=1}^{46}b_i\right)}{b_i\left(\sum_{i=1}^{46}b_i\right)}\right)=\frac{\sum_{i=1}^{46}a_i}{\sum_{i=1}^{46}b_i}$$

$\therefore$

$$\sum_{i=1}^{46}\left[\left(A_i\frac{\left(\sum_{i=1}^{46}b_i\right)}{b_i}\right)\frac{a_i}{\left(\sum_{i=1}^{46}b_i\right)}\right]=\frac{\sum_{i=1}^{46}a_i}{\sum_{i=1}^{46}b_i}$$

$\because \sum_{i=1}^{46}b_i$ is independent on i,

$\therefore$

$$\sum_{i=1}^{46}\left[\left(A_i\frac{\sum_{i=1}^{46}b_i}{b_i}\right)a_i\right]=\sum_{i=1}^{46}a_i$$

As a result, in order to satisfy Eq. B.2, A_i should be $\frac{b_i}{\sum_{i=1}^{46}b_i}$ which is used to define $P_b(q_i)$, the probability of phoneme q_i:

$$P_b(q_i)=\frac{\#((*,*,q_i)\text{ in the utterances of all backgroud speakers})}{\#((*,*,*)\text{ in the utterances of all backgroud speakers})}=\frac{b_i}{\sum_{i=1}^{46}b_i}. \quad \text{(B.3)}$$

Similarly, we can also prove that

$$\boxed{P_b(m,p|q) = \sum_{k=1}^{M} [P_{s_k}(m,p|q)P(s_k)]}$$

where M is the total number of background speakers used for training the background models, s_k is one of these background speakers, and $P(s_k)$ is the probability of speaker s_k:

$$P(s_k) = \frac{\#((*,*,q) \text{ in the utterances of backgroud speaker } s_k)}{\#((*,*,q) \text{ in the utterances of all backgroud speakers})},$$

and

$$\boxed{P_s(m,p|*) = \sum_{i=1}^{46} [P_s(m,p|q_i)P_s(q_i)]}\ .$$

As a result, we can derive:

$$\boxed{\frac{P_b(m,p|q)}{P_b(m,p|*)}P_s(m,p|*) = \frac{\left[\sum_{k=1}^{M} P_{s_k}(m,p|q)P(s_k)\right] \cdot \left[\sum_{i=1}^{46} P_s(m,p|q_i)P_s(q_i)\right]}{\sum_{k=1}^{M}\sum_{i=1}^{46} P_{s_k}(m,p|q_i)P(s_k)P_b(q_i)}}$$

Appendix C

AUTHOR'S PUBLICATIONS

International Journal Papers

1. S.X. Zhang, M.W. Mak and Helen M. Meng, "Speaker Verification via High-Level Feature Based Phonetic-Class Pronunciation Modeling", *IEEE Trans. on Computers*, Vol. 56, No. 9, pp. 1189-1198, Sept. 2007.

2. S.X. Zhang, M.W. Mak, "A New Adaptation Approach to High-Level Speaker-Model Creation in Speaker Verification", *Speech Communications*, *submitted* .

International Conference Papers

1. S.X. Zhang, M. W. Mak, and H. Meng, "High-Level Feature-Based Speaker Verification via Articulatory Phonetic-Class Pronunciation Modeling", *Interspeech'2007*, pp. 762-765, Antwerp, Belgium.

2. S.X. Zhang and M. W. Mak, "A New Adaptation Method for Speaker-Model Creation in High-Level Speaker Verification", *Advances in Multimedia Information Processing (PCM'2007)*, Hong Kong, Springer LNCS 4810, pp. 325-335.

3. S.X. Zhang and M. W. Mak, "Articulatory-Feature based Sequence Kernel For High-Level Speaker Verification", *2008 International Conference on Machine Learning and Cybernetics (ICMLC'2008)*.

4. S.X. Zhang and M. W. Mak, "High-Level Speaker Verification via Articulatory-Feature based Sequence Kernels and SVM", *Interspeech'2008*, *accepted*.