Bayesian Distance Metric Learning on i-vector for Speaker Verification

by

Xiao Fang

B. S., Electrical Engineering
University of Science and Technology of China, 2011

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Author ....
Department of Electrical Engineering and Computer Science
August 20, 2013

Certified by ....
James. R. Glass
Senior Research Scientist
Thesis Supervisor

Certified by ....
Najim Dehak
Research Scientist
Thesis Supervisor

Accepted by ....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Bayesian Distance Metric Learning on i-vector for Speaker Verification

by

Xiao Fang

Submitted to the Department of Electrical Engineering and Computer Science
on August 20, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

This thesis explores the use of Bayesian distance metric learning (Bayes_dml) for the task of speaker verification using the i-vector feature representation. We propose a framework that explores the distance constraints between i-vector pairs from the same speaker and different speakers. With an approximation of the distance metric as a weighted covariance matrix of the top eigenvectors from the data covariance matrix, variational inference is used to estimate a posterior distribution of the distance metric. Given speaker labels, we select different-speaker data pairs with the highest cosine scores to form a different-speaker constraint set. This set captures the most discriminative between-speaker variability that exists in the training data. This system is evaluated on the female part of the 2008 NIST SRE dataset. Cosine similarity scoring, as the state-of-the-art approach, is compared to Bayes_dml. Experimental results show the comparable performance between Bayes_dml and cosine similarity scoring. Furthermore, Bayes_dml is insensitive to score normalization, as compared to cosine similarity scoring. Without the requirement of the number of labeled examples, Bayes_dml performs better in the context of limited training data.

Thesis Supervisor: James. R. Glass
Title: Senior Research Scientist

Thesis Supervisor: Najim Dehak
Title: Research Scientist

Acknowledgments

First and foremost, I would like to thank Jim Glass and Najim Dehak for offering me the opportunity to do research in their group. I am grateful to Jim for his consideration and patience all the time, for always guiding me down the right path. I appreciate Najim's passion and brilliance. Najim's broad knowledge and thoughtful insights in this field have inspired me a lot through this thesis work. This thesis would not have been possible without them.

The Spoken Language Systems group has provided me a research home in the past two years. Thank you to all the members for your energy, creativity, and friendship. The birthday celebrations, the spectrum reading seminars, and the defense suit-ups would be unforgettable memories in my life.

I would like to thank all my friends, old and new, locally available and geographically separated, for accompanying and encouraging me, for sharing tear and joy with me. Lastly I would like to thank my parents and my elder brother for their love, inspiration, and support all these years.

This thesis was supported by DARPA under the RATS (Robust Automatic Transcription of Speech) Program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speaker verification is the use of a machine to verify a person's claimed identity from his/her voice. The applications of speaker verification cover almost all the areas where it is necessary to secure actions, transactions, or any type of interactions by identifying the person. Currently, most applications are in the banking and telecommunication areas. Compared to other biometric systems, which are based on different modalities, such as a fingerprint or face image, the voice has some compelling advantages [1]. First, speech is easy to get at low cost. The telephone system provides a ubiquitous approach to obtain and deliver speech signals. For telephone-based applications, there is no need to install special signal transducers or networks at application access points since a cell phone gives one access almost everywhere. For non-telephone applications, sound cards and microphones are also cheap devices that are readily available. Second, speech is a natural signal which is not considered threatening by users. Users won't consider providing a speech sample for authentication as an intrusive step.

In the last decade, research in speaker verification has made great improvements and we have seen successful commercial applications in some products. Depending on whether the spoken phrase is fixed or not, a speaker verification system can be classified as text-dependent or text-independent [1]. We focus on the text-independent speaker verification system in this research.

A typical speaker verification system involves two steps: feature extraction from the speech signal, and statistical modeling of feature parameters. Since it was proposed in the mid 1990s, Gaussian Mixture Models (GMMs) have become the dominant approach for modeling text-independent speaker verification [2]. In the past decade, the GMM-based system with Bayesian adaptation of speaker models from a universal background model and score normalization has achieved the top performance in the NIST (National Institute of Standards and Technology) speaker recognition evaluations (SRE) [3]. This system is referred to as the Gaussian Mixture Model-Universal Background Model (GMM-UBM) speaker verification system.

In the GMM-UBM approach, the speaker's model is derived from the UBM via a *maximum a posterior* (MAP) adaptation. When the speaker training data is limited, some Gaussian components were prevented from being adapted [10]. In order to address this problem, the theory of Joint Factor Analysis (JFA) is used for speaker modeling [6]. JFA-based methods model both speaker and channel/session variability in the context of a GMM [4] [5]. A more recent approach represents all the variabilities in a single low-dimensional space named total variability space, with no distinction between speaker and channel subspaces [13]. A speech utterance is represented by a new vector called total factors (also referred to as an i-vector) in this new space. The i-vector contains both speaker- and channel-variability. We can generally treat i-vectors as input to common classifiers such as Support Vector Machines (SVMs), a cosine distance classifier, or probabilistic linear discriminant analysis (PLDA). In [13], the authors show that cosine distance scoring achieves state-of-the-art performance. In the i-vector training and score verification process, we don't use speaker labels at all, which suggests that algorithms with the full use of speaker labels might get better performance.

Note that the basic speaker verification task is to determine whether the test utterance and the target utterance are from the same speaker. Thus we can view the speaker verification system as a distance metric leaning problem: given speaker labels of training utterances, we aim to find an appropriate distance metric that brings "similar" utterances (belonging to the same speaker) close together while separating "dissimilar" utterances (belonging to different speakers) [32]. In this thesis, we present a speaker verification system based on the distance metric learning framework. In [33], Yang and Jin present a Bayesian framework for distance metric learning, which has achieved high classification accuracy in image classification. In addition, this approach is insensitive to the number of labeled examples for each class, as compared to most algorithms requiring a large number of labeled examples [33]. This advantage is particularly important for realistic speaker verification systems, as it can be difficult to collect plenty of samples from every speaker in many industrial applications, although possible to collect samples from a large number of different speakers.

The rest of this thesis is organized as follows: Chapter 2 will give a background review of speech parameterization and Gaussian Mixture Models. Chapter 3 will introduce the theory of factor analysis in speaker verification. The compensation techniques to remove the nuisance variabilities among different trials are explained afterwards in Chapter 4. Then, the Bayesian distance metric learning framework is presented in Chapter 5. Chapter 6 will provide the experimental set up for the system, and show some results. Finally, Chapter 7 concludes this thesis and suggests possible directions for future work.

Chapter 2
Background and Related Work

The speech signal conveys rich information, such as the words or message being spoken, the language spoken, the topic of the conversation, and the emotion, gender and identity of the speaker. Automatic speaker recognition aims to recognize the identity of the speaker from a person's voice. The general area of speaker recognition involves two fundamental tasks. The *Speaker Identification* task is to determine who produces the speech test segment. Usually it is assumed that the unknown voice must come from a fixed set of known speakers. Thus, the system performs a $1 : N$ classification, referred to as a *closed-set* identification. The *Speaker Verification* task is to determine whether the claimed identity of the speaker is the same as the identity of the person who produced the speech segment. In other words, given a segment of speech and a hypothesized speaker Q, the task of speaker verification is to determine if this segment was spoken by the speaker Q. Since the impostors who falsely claim to be a target speaker are generally not known to the system, this task is referred to as an *open-set* classification. This thesis studies the problem of *Speaker Verification.*

In most speaker verification systems, an input speech utterance is compared to an enrolled *target* speaker model, resulting in a similarity measure computed between them, also called a *similarity score.* The process of computing a score from a speaker model and a test speech utterance is usually called a *trial.* The *trials* may be classified as *target* and *non-target* trials depending on whether the training and test speech are respectively generated by the same individual or not. The users attempting to access the system are referred to as *target* users when their identity is the same as the claimed one, otherwise they are called *impostors.*

Human speech contains numerous discriminative features that can be used to identify speakers. The objective of automatic speaker verification is to extract, characterize, and recognize the information about speaker identity. The speech signal is first transformed to a set of feature vectors in a front-end processing step. The aim of this transformation is to obtain a new representation that is more compact, less redundant, and more suitable for statistical modeling. The output of this stage is typically a sequence of feature vectors $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$, where x_l is a feature vector indexed at an index $l \in \{1, 2, \ldots, L\}$.

An implicit assumption often used is that $\mathbf{x}$ contains speech from only one speaker. Thus, this task is better termed *single speaker verification*. The *single speaker verification* task can be stated as a basic hypothesis test between two hypotheses:

H_0: $\mathbf{x}$ is from the hypothesized speaker Q,

H_1: $\mathbf{x}$ is *not* from the hypothesized speaker Q.

The optimum test to decide between these two hypotheses is to apply the likelihood ratio test given by

$$\frac{P(\mathbf{x}|H_0)}{P(\mathbf{x}|H_1)} = \begin{cases} > \beta & \text{accept } H_0 \\ < \beta & \text{accept } H_1 \end{cases}$$

where β is the decision threshold. Since the likelihood is usually very small and may exceed the maximum precision, it is often to use log likelihood ratio instead.

$$\log \frac{P(\mathbf{x}|H_0)}{P(\mathbf{x}|H_1)} = \begin{cases} > \beta & \text{accept } H_0 \\ < \beta & \text{accept } H_1 \end{cases}$$

The main goal in designing a speaker detection system is to determine techniques to compute values for the two likelihoods, $P(\mathbf{x}|H_0)$ and $P(\mathbf{x}|H_1)$.

This chapter will introduce the commonly used speech parametrization techniques and the statistical modeling to calculate the likelihoods.

■ 2.1 Speech Parameterization

Most current speech parameterizations used in speaker verification systems rely on a cepstral representation of speech [1]. The extraction and selection of the best parametric representation of acoustic signals is an important task in the design of any speaker verification systems [36]. Some of the audio features that have been successfully used in the field include Mel-frequency cepstral coefficients (MFCC), Linear predictive coding (LPC), etc. The most popular is MFCC, which is the result of the cosine transform of the real logarithm of the short-term energy spectrum expressed on a mel-frequency scale [36]. The calculation of the MFCC includes the following steps.

A. *Mel-frequency warping*

The human perception of sound frequency does not follow a linear scale. For each tone with an actual frequency, f, measured in Hz, a subjective pitch is measured on a scale called the mel scale, which is a perceptual scale of pitches judged by listeners to be equal in distance from one another [41]. The mel-frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000Hz. As a reference point, the pitch of a 1000 Hz tone, 40dB above the perceptual hearing threshold, is defined as 1000 mels. Therefore we can use the following approximate formula to compute the mel frequency for a given frequency f in Hz.

$$\mathrm{Mel}(f) = 2595 \times log_{10}(1 + \frac{f}{700}) \tag{2.1}$$

The common approach to approximate the subjective spectrum is to use a filter bank. The speech signal is first sent to a high-pass filter to compensate the high-frequency part that was suppressed during the sound production and to amplify the the importance of high-frequency formants, and then segmented into frames. Each frame is multiplied with a hamming window in order to keep the continuity of the boundary. We perform a discrete Fourier transform on each frame and transform them to the mel-frequency spectrum via the filter bank. The filter bank has a triangular band pass frequency response, and the center frequency spacing and the bandwidth are determined by a constant mel-frequency interval. The mel scale filter bank used in this thesis is a series of 23 triangular band pass filters that have been designed to approximate the band pass filtering believed to occur in the auditory system. The log energy within each filter is log mel-frequency spectral coefficient, denoted as S_j, $j = 1, 2, ..., 23$.

B. *Cepstrum*

In the final step, we convert the log mel spectrum back to “time”. The result is called the Mel Frequency Cepstral Coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the mel-frequency spectral coefficients (and their logarithms) are real numbers, we can convert them to time-like domain, called quefrency domain, using the discrete cosine transform (DCT).

$$C_i = \sum_{j=1}^{23} S_j \cdot cos[\frac{\pi \cdot i}{23}(j - \frac{1}{2})] \tag{2.2}$$

The complete process for the calculation of MFCC is shown in Figure 2.1.

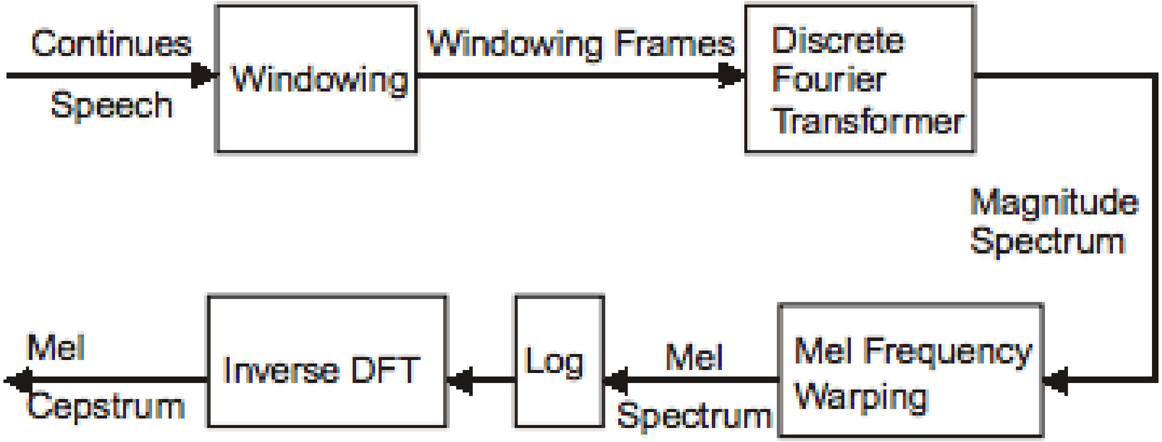

Figure 2.1: *Pipeline for MFCC*

■ 2.2 The GMM-UBM Approach

The next step after obtaining the parametrization representation is the selection of the likelihood function $P(\mathbf{x}|H_0)$ and $P(\mathbf{x}|H_1)$. For notational purposes, we can let H_0 be represented by a probabilistic model λ_Q that characterizes the hypothesized speaker Q, and we can use $\lambda_{\overline{Q}}$ to represent the probabilistic model of the alternative hypothesis H_1. The classical approach is to model each speaker as a probabilistic source with unknown but fixed probability density function. While the model in λ_Q is well defined and can usually be estimated via some enrollment speech from the speaker Q, the model for $\lambda_{\overline{Q}}$ is less well defined since it potentially must represent the entire space of possible alternatives to the hypothesized speaker. The approach typically used to tackle the problem of alternative hypothesis modeling is to pool speech from many non-target speakers and train a single model known as the Universal Background Model (UBM). The advantage of this approach is that a single speaker-independent model can be trained once for a particular task and then used for all hypothesized speakers in the task [9].

The GMM is a generative model used widely in speaker verification to model the feature distribution. A GMM is composed of a finite mixture of multivariate Gaussian components. Given a GMM θ consisting of C components, the likelihood of observing an F-dimensional feature vector x is defined as

$$P(x|\theta) = \sum_{c=1}^{C} \pi_c N_c(x|\mu_c, \Sigma_c) \tag{2.3}$$

where the mixture weights $\pi_c \geq 0$ are constrained by $\sum_c \pi_c = 1$, and $N_c(x|\mu_c, \Sigma_c)$ is a multivariate Gaussian with F-dimensional mean vector, μ_c, and $F \times F$ covariance matrix, Σ_c.

$$N_c(x|\mu_c,\, \Sigma_c) = \frac{1}{(2\pi)^{2F}|\Sigma_c|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1}(x-\mu_c)\} \tag{2.4}$$

The parameters of the model are denoted as $\theta = \{\theta_1, \theta_2, \ldots, \theta_C\}$, where $\theta_c = \{\pi_c, \mu_c, \Sigma_c\}$.

While the general model form supports full covariance matrices, typically only diagonal covariance matrices are used. This is because the density modeling of an M-th order full covariance GMM can generally be equally achieved using a larger-order diagonal covariance GMM and diagonal covariance GMMs are more computationally efficient than full covariance GMMs [1].

For a sequence of feature vectors $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$, we assume that each observation vector is independent of the others. The likelihood of the given utterance $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$ is the product of the likelihood of each of the L frames. The log likelihood is computed as

$$\log P(\mathbf{x}|\theta) = \sum_{t=1}^{L} \log P(x_l|\theta) \tag{2.5}$$

The maximum likelihood (ML) parameters of a model θ can be estimated via the Expectation-Maximization (EM) algorithm. The equations for the ML parameter updates can be found in [8]. The UBM is trained on a selection of speech that is reflective of the expected alternative speech to be encountered during recognition. It represents the speaker-independent distribution of features.

For the speaker model, a single GMM can be trained on the speaker's enrollment data, however, the amount of speaker-specific data would be much too limited to give a good representation of the speaker. We may end up modeling the channel characteristics or other aspects of the data instead. In contrast, the larger abundance of speech data used to estimate the UBM might be a better starting point for modeling a specific speaker. Thus we derive the speaker's model via a *maximum a posterior* (MAP) adaptation from the well-trained parameters in the UBM. This provides a tighter coupling between the speaker's model and the UBM, which not only produces better performance than separate (decoupled) models, but also allows for a fast-scoring technique.

The MAP adaptation is similar to the EM algorithm and it also allows the fast log-likelihood ratio scoring technique. Given a UBM parameterized by θ_{UBM} and training feature vectors from a speaker $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$, we first calculate the probabilistic alignment between each training frame and the UBM mixture components. For UBM mixture c, we compute

$$\gamma_l(c) = P(c|x_l, \theta_{UBM}) = \frac{\pi_c N_c(x_l|\mu_c, \Sigma_c)}{\sum_{c=1}^{C} \pi_c N_c(x_l|\mu_c, \Sigma_c)} \tag{2.6}$$

and the relevant Baum-Welch statistics for the weight, mean, and covariance parameters of the UBM are:

$$N_c(\mathbf{x}) = \sum_{l=1}^{L} P(c|x_l, \theta_{UBM}) = \sum_{l=1}^{L} \gamma_l(c) \tag{2.7}$$

$$\bar{F}_c(\mathbf{x}) = \frac{1}{N_c(\mathbf{x})} \sum_{l=1}^{L} \gamma_l(c) \cdot x_l \tag{2.8}$$

$$\bar{S}_c(\mathbf{x}) = \frac{1}{N_c(\mathbf{x})} \sum_{l=1}^{L} \gamma_l(c) \cdot x_l x_l^* \tag{2.9}$$

The UBM sufficient statistics for mixture c are updated from these sufficient statistics of the training data to generate adapted parameters as below:

$$\hat{\pi}_c = \beta \left(\alpha_c \frac{N_c(\mathbf{x})}{L} + (1 - \alpha_c)\pi_c \right) \tag{2.10}$$

$$\hat{\mu}_c = \alpha_c \bar{F}_c(\mathbf{x}) + (1 - \alpha_c)\mu_c \tag{2.11}$$

$$\hat{\Sigma}_c = \alpha_c \bar{F}_c(\mathbf{x}) + (1 - \alpha_c)(\Sigma_c + \mu_c \mu_c^*) - \hat{\mu}_c \hat{\mu}_c^* \tag{2.12}$$

β is a scale factor computed over all adapted mixture weights to ensure that $\sum_c \hat{\pi}_c = 1$, and α_c are the data-dependent adaptation coefficients controlling the balance between old and new estimates of the GMM parameters. The coefficients are defined as

$$\alpha_c = \frac{N_c(\mathbf{x})}{N_c(\mathbf{x}) + r} \tag{2.13}$$

where r is a constant relevance factor.

The data-dependent adaptation coefficient allows mixture-dependent adaptation of parameters. For mixture components with a low probabilistic count $N_c(\mathbf{x})$ of the user data, $\alpha_c \to 0$ will cause the deemphasis of the new parameters and the emphasis of the old parameters. For mixture components with a high probabilistic count $N_c(\mathbf{x})$, $\alpha_c \to 1$ will cause the use of the new speaker-dependent parameters. The relevance factor controls how much new data should be observed in a mixture when updating the old parameters with the new parameters. Thus this approach should be robust to limited training data.

The adaptation of the mean and covariance parameters of the observed Gaussians is displayed in Figure 2.2. In practice, only the mean vectors μ_c, $c = 1, ..., C$, are adapted, while updated weights and covariance matrices do not significantly affect system performance. The selection of the number of Gaussian components depends on the type and the amount of training data, such as telephone data or microphone data, gender-independent or gender-dependent.

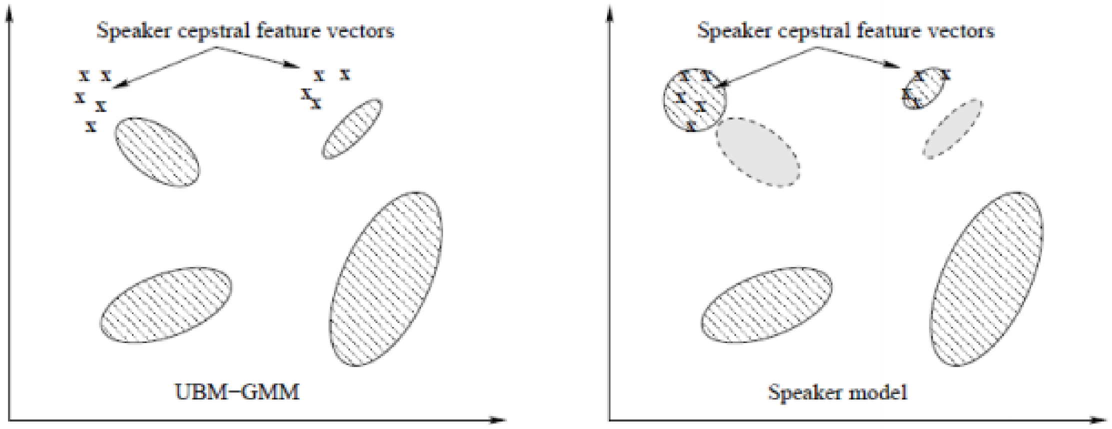
Speaker cepstral feature vectors
UBM−GMM
Speaker cepstral feature vectors
Speaker model

Figure 2.2: *maximum a posteriori (MAP) adaptation [3] [9]*

■ 2.3 Data Sets and Evaluations

Our experiments are carried out on the NIST 2008 speaker recognition evaluation (SRE) dataset [39]. NIST SRE is an ongoing series of evaluations to focus on the core technology issues in the field of text independent speaker recognition. The systems have to answer the question, "Did speaker X produce the speech recording Y and to what degree?". Each trial requires a decision score to reflect the system's estimate of the probability that the test segment contains speech from the target speaker.

Detection system performance is usually characterized in terms of two error measures, namely miss probability $P_{Miss/Target}$ and false alarm $P_{FalseAlarm/Nontarget}$. These respectively correspond to the probability of not detecting the target speaker when present, and the probability of falsely detecting the target speaker when not present. Different operating points will generate different $P_{Miss/Target}$ and $P_{FalseAlarm/Nontarget}$. We care more about the operating point where the two error rates are equal, and the resulting rate is called equal error rate (EER). Another formal evaluation measure is the detection cost function (DCF), defined as a weighted sum of the miss probability and false alarm:

$$C_{Det} = C_{Miss} \times P_{Miss/Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm/Nontarget} \times (1 - P_{Target}) \quad (2.14)$$

The parameters are C_{Miss} and $C_{FalseAlarm}$, the relative cost of detection errors, and P_{Target}, the a priori probability of the specified target speaker. The primary evaluation will use $C_{Miss} = 10$, $C_{FalseAlarm} = 1$, and $P_{Target} = 0.01$.

In addition to the single number measures of minDCF and EER, more information can be shown in a graph plotting all the operating points. An individual operating point corresponds to a score threshold for separating actual decisions of true or false. All possible system operating points are generated by sweeping over all possible threshold values. NIST has introduced Decision Error Tradeoff (DET) Curves since the 1996 evaluation [3], where the two error rates are plotted on the x and y axes on a normal deviate scale on the receiver operating characteristic (ROC) curve. The DET Curves have been widely used to represent the detection system performance. The C_{Det} value and EER correspond to a specific operating point on the DET curve.

■ 2.4 Chapter Summary

In this chapter, we have described the speech parameterization to transform a speech utterance to a sequence of feature vectors for statistical modeling. Our focus was on the computation of MFCCs, since they are used in subsequent chapters. We have also presented the GMM-UBM approach, the classical statistical modeling approach for speaker recognition. The *maximum a posterior* approach to obtain the speaker model from the UBM is fully dealt with and the benefit of this adaption is explained. Finally the datasets and evaluation metric for experiments were introduced.

Chapter 3
Factor Analysis Based Speaker Verification

The GMM-UBM approach achieved great success, but suffered from data sparsity in MAP adaptation [9]. Since each Gaussian component is updated independently, some components of the UBM were prevented from being adapted, and thus failed to capture the thorough and complete representation of the speaker's true model in the presence of limited speaker training data [6]. It is necessary to correlate or link together the different Gaussian components of the UBM. The theory of Joint Factor Analysis (JFA) is used to achieve this goal [25].

This chapter will present a thorough description of the idea and mechanism of Joint Factor Analysis. A good overview of JFA for speech processing can also be found in [9]. Two scoring approaches, cosine similarity scoring and probabilistic linear discriminant analysis, are introduced afterwards.

■ 3.1 Joint Factor Analysis

In the JFA framework, a speaker model obtained by adapting from a UBM (parameterized with C mixture components in a feature space of dimension F) can also be viewed as a single supervector of dimension $C \cdot F$ along with a diagonal super-covariance matrix of dimension $CF \times CF$ [7] [9]. The supervector is generated by concatenating the mean vector of each Gaussian mixture, while the super-covariance matrix is generated by concatenating the diagonal covariance matrix of each mixture along its diagonal.

The idea behind factor analysis is that a measured high-dimensional vector, i.e. speaker supervector, may be believed to lie in a lower-dimensional subspace. Another assumption in JFA is that the speaker- and channel-dependent supervector M for a given utterance can be broken down into the sum of two supervectors

$$M = s + c \tag{3.1}$$

where the supervector s depends on the speaker, and the supervector c depends on the channel. They can be modeled as

$$s = m + Vy + Dz \tag{3.2}$$

$$c = Ux \tag{3.3}$$

where m is the speaker- and channel-independent supervector interpreted as the initial UBM supervector. V and U are low-rank matrices that represent the lower dimensional subspaces in which the speakers and channels lie, known as the eigenvoices and the eigenchannels, respectively. Lastly, D is a diagonal $CF \times CF$ matrix to model the residual variabilities of the speakers not captured by V. The vectors y, z and x are the speaker- and session-dependent factors in their respective subspaces, and each is assumed to be a random variable with a normal distribution $N(0, I)$. The basic idea is displayed in Figure 3.1 and a detailed explanation can be found in [8]. The fact that the three latent variables y, z, and x are estimated jointly accounts for the terminology *Joint* Factor Analysis.

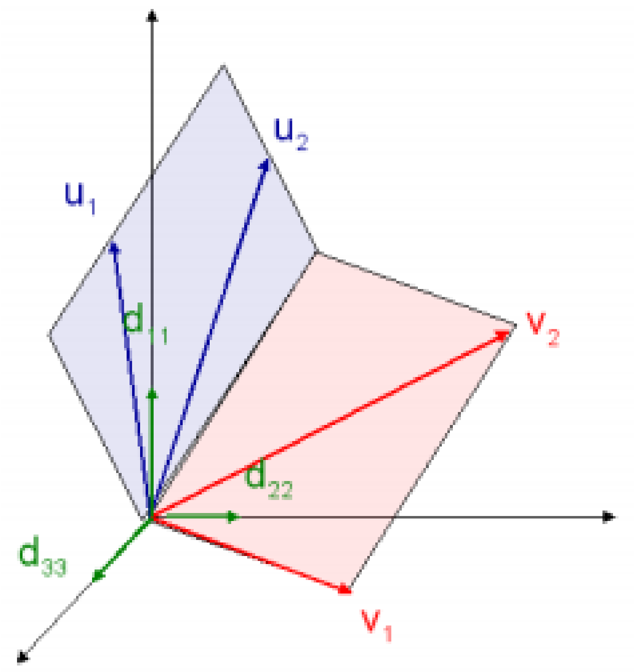

Figure 3.1: *essentials of Joint Factor Analysis [7] [9]*

The JFA approach represents speaker variabilities, and compensates for channel variabilities better than GMM-UBM approach, while it is complex in both theory and implementation. A simplified solution, called total variability, was subsequently developed with superior performance.

■ 3.2 Total Variability Approach

Experiments show that the channel factors in Joint Factor Analysis also contain information about speakers [8]. Based on this, an approach was proposed that does not distinguish between speaker variability and channel variability. Given an utterance, the speaker- and channel-dependent GMM supervector M can be represented as

$$M = m + Tw \tag{3.4}$$

where m is the speaker- and channel-independent supervector (which can be taken to be the UBM supervector), T is a rectangular matrix of low rank and w is a random vector having standard normal distribution $N(0, I)$. T defines the new total variability space and the remaining variabilities not captured by T are accounted for in a diagonal covariance matrix Σ. In this model, the high-dimensional supervectors lie around m in a relatively lower-dimensional subspace and w is the speaker- and channel-dependent factor in the total variability space. The mean of the posterior distribution of w corresponds to a total factor vector, or an i-vector, which can be seen as a low dimensional speaker verification feature. The i-vector is short for Intermediate Vector, for the intermediate representation between an acoustic feature vector and a supervector, or Identity Vector, for its compact representation of a speaker's identity [13].

The parameter training in the total variability approach is also based on the EM algorithm [13] [31]. The main difference from learning the eigenvoice V is that each recording of a given speaker's set of utterances is regarded as having been produced by a different speaker in training T, whereas all the utterances of a given speaker are considered to belong to the same person in training V. The speaker characteristics are not learned explicitly in the total variability approach, while the latent variable y represents the speaker variability in Joint Factor Analysis. A thorough explanation of the key details for estimating T and extracting w can be found in [9].

From here on we will use the posterior mean of w as a low-dimensional representation of the utterance. The fixed-length i-vectors can be used as input to standard recognition algorithms to produce the desired likelihood score [11] [26]. Two scoring approaches are introduced next: cosine similarity scoring and probabilistic linear discriminant analysis.

■ 3.3 Cosine Similarity Scoring

As the only latent variable learned from each utterance, the low-dimensional i-vector is a full and final representation of a speaker's and channel's identity. Thus, total variability can be used as a front end feature extraction method, and there is no need to calculate the log-likelihood ratio scoring function like the GMM-UBM and JFA approaches [26]. Recently, cosine similarity scoring has been applied to compare two i-vectors for making a speaker detection decision [13]. With i-vectors of the target speaker utterance w_{target} and the test speaker utterance w_{test} in hand, the verification is carried out using the cosine similarity score as below:

$$score(w_{target}, w_{test}) = \frac{w_{target}^t \cdot w_{test}}{\|w_{target}\| \cdot \|w_{test}\|} \gtrless \beta \tag{3.5}$$

where β is the decision threshold.

Since the i-vector contains both the speaker and session variabilities, we need to do session compensation for cosine similarity scoring, which will be explained in detail in Section 4.1. A more sophisticated approach to directly model session variability within i-vectors was recently introduced by Kenny [17] [20] as Probabilistic Linear Discriminant Analysis (PLDA) [18].

■ 3.4 Probabilistic Linear Discriminant Analysis

Probabilistic Linear Discriminant Analysis (PLDA) is similar to the JFA approach, but uses i-vectors rather than GMM supervectors as the basis for factor modeling [18]. Suppose there are I speakers each of J utterances in the training set. The jth i-vector of the ith speaker is denoted by w_{ij}. We model data generation by the process

$$w_{ij} = m + \mathbf{F}\mathbf{s}_i + \mathbf{G}\mathbf{u}_{i,j} + \epsilon_{i,j} \tag{3.6}$$

Each utterance is comprised of two parts: the signal component $m + \mathbf{F}\mathbf{s}_i$ which only depends on the speaker identity but not on the particular utterance; and the noise component $\mathbf{G}\mathbf{u}_{i,j} + \epsilon_{i,j}$ which is different for every utterance of the speaker and represents session variability. In Equation 3.6, m is the overall mean of all the training utterances. $\mathbf{F}$ is the eigenvoice matrix and $\mathbf{G}$ is the eigenchannel matrix. The columns of $\mathbf{F}$ and $\mathbf{G}$ contain the basis for the between-speaker subspace and within-speaker subspace, respectively. And $\mathbf{s}_i$ and $\mathbf{u}_{i,j}$ represent the position in the corresponding subspace. The remaining variability not captured is explained by the residual noise term $\epsilon_{i,j}$ following a Gaussian prior with diagonal covariance Σ. Usually we define Gaussian priors on the latent variables $\mathbf{s}_i$ and $\mathbf{u}_{i,j}$. Kenny [17] investigated using both Gaussian and heavy-tailed prior distributions for $\mathbf{s}_i$, $\mathbf{u}_{i,j}$, and $\epsilon_{i,j}$, but we only investigate the Gaussian priors. This model can also be described in terms of the following conditional probabilities

$$P(w_{ij}|\mathbf{s}_i, \mathbf{u}_{ij}, \theta) = N(m + \mathbf{F}\mathbf{s}_i + \mathbf{G}\mathbf{u}_{ij}, \Sigma) \tag{3.7}$$

$$P(\mathbf{s}_i) = N(0, \mathbf{I}) \tag{3.8}$$

$$P(\mathbf{u}_i) = N(0, \mathbf{I}) \tag{3.9}$$

Using this model involves two steps: the training phase to learn the parameters $\theta = \{m, \mathbf{F}, \mathbf{G}, \Sigma\}$; and the recognition phase to make inferences whether two utterances come from the same speaker. The latent variable $\mathbf{s}_i$ identifies the speaker. Thus recognition is conducted to evaluate the likelihood that two utterances are generated from the same underlying $\mathbf{s}_i$.

■ 3.4.1 Training

The parameters $\theta = \{m, \mathbf{F}, \mathbf{G}, \Sigma\}$ are obtained to maximize the likelihood of the training dataset. Similar to the problem in the total variability approach, latent variables and parameters are both unknown and need to be estimated. We can also use the EM algorithm to estimate the two sets of parameters.

- E-step: Calculate the full posterior distribution over the latent variables $\mathbf{s}_i$ and $\mathbf{u}_{i,j}$, given the parameter values. We simultaneously estimate the joint probability distribution of all the latent variable $\mathbf{s}_i$, $\mathbf{u}_{i1,\ldots,iJ}$ that pertain to each speaker. First we combine the generative equations for all of the N utterances as follows

$$\begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{iN} \end{bmatrix} = \begin{bmatrix} m \\ m \\ \vdots \\ m \end{bmatrix} + \begin{bmatrix} \mathbf{F} & \mathbf{G} & \mathbf{F} & \cdots & 0 \\ \mathbf{F} & 0 & \mathbf{G} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F} & 0 & 0 & \cdots & \mathbf{G} \end{bmatrix} + \begin{bmatrix} \mathbf{s}_i \\ \mathbf{u}_{i1} \\ \mathbf{u}_{i2} \\ \vdots \\ \mathbf{u}_{iN} \end{bmatrix} + \begin{bmatrix} \epsilon_{i1} \\ \epsilon_{i2} \\ \vdots \\ \epsilon_{iN} \end{bmatrix} \tag{3.10}$$

We rename these composite matrices as

$$w_i = m' + \mathbf{A}\mathbf{y}_i + \epsilon_i \tag{3.11}$$

This compound model is rewritten in terms of probabilities

$$P(w_i|\mathbf{y}_i) = N(\mathbf{A}\mathbf{y}_i, \Sigma') \tag{3.12}$$

$$P(\mathbf{y}_i) = N(0, \mathbf{I}) \tag{3.13}$$

where

$$\Sigma' = \begin{bmatrix} \Sigma & 0 & \cdots & 0 \\ 0 & \Sigma & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma \end{bmatrix} \tag{3.14}$$

Applying Bayes' rule, we obtain the posterior distribution as

$$P(\mathbf{y}_i|w_i,\theta) \propto P(w_i|\mathbf{y}_i,\theta)P(\mathbf{y}_i) \tag{3.15}$$

The posterior on the left must be Gaussian since both terms on the right are Gaussians. It can be shown that the first two moments of this Gaussian are

$$\mathbb{E}[\mathbf{y}_i] = (\mathbf{A}^T\Sigma^{'-1}\mathbf{A}+\mathbf{I})^{-1}\mathbf{A}^T\Sigma^{'-1}(w_i - m^{'}) \tag{3.16}$$

$$\mathbb{E}[\mathbf{y}_i\mathbf{y}_i^T] = (\mathbf{A}^T\Sigma^{'-1}\mathbf{A}+\mathbf{I})^{-1} + \mathbb{E}[\mathbf{y}_i]\mathbb{E}[\mathbf{y}_i]^T \tag{3.17}$$

- M-step: Optimize the point estimates of the parameters $\theta = \{m, \mathbf{F}, \mathbf{G}, \Sigma\}$. We rewrite Equation 3.6 as

$$\begin{aligned} w_{ij} &= m + \begin{bmatrix} \mathbf{F} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{s}_i \\ \mathbf{u}_{ij} \end{bmatrix} + \epsilon_{ij} \\ &= m + \qquad \mathbf{B} \qquad\quad \mathbf{z}_{ij} \quad + \epsilon_{ij} \end{aligned} \tag{3.18}$$

The M-step aims to optimize

$$Q(\theta_t,\theta_{t-1}) = \sum_i \sum_j \int P(\mathbf{z}_i|w_{i1},...,w_{iJ},\theta_{t-1})\log\left[P(w_{ij}|\mathbf{z}_{ij})P(\mathbf{z}_i)\right]d\mathbf{z}_i \tag{3.19}$$

where t is the iteration index. We take the derivatives with respect to $\mathbf{B}$ and Σ and equal them to zero to obtain the update rules

$$m = \frac{1}{IJ}\sum_{ij} w_{ij} \tag{3.20}$$

$$\mathbf{B} = \left(\sum_{i,j}(w_{ij}-m)\mathbb{E}[\mathbf{z}_i]^T\right)\left(\sum_{i,j}\mathbb{E}[\mathbf{z}_i\mathbf{z}_i^T]\right)^{-1} \tag{3.21}$$

$$\mathbf{\Sigma} = \frac{1}{IJ}\sum_{i,j}\text{diag}\ \left[(w_{ij}-m)(w_{ij}-m)^T - \mathbf{B}\mathbb{E}[\mathbf{z}_i](w_{ij}-m)^T\right] \tag{3.22}$$

The expectation terms $\mathbb{E}[\mathbf{z}_i]$ and $\mathbb{E}[\mathbf{z}_i]$ can be generated from Equation 3.16 and 3.17, and the equivalence between $\mathbf{y}_i$ and $\mathbf{z}_i$. The updated rules of $\mathbf{F}$ and $\mathbf{G}$ can be retrieved from $\mathbf{B}$ according to the equivalence from Equation 3.18.

■ 3.4.2 Recognition

Given two i-vectors w_{target} and w_{test}, the similarity score can be computed as the logarithm of the ratio of the of the two hypothesis: H_0, both w_{target} and w_{test} belong to the same speaker (same $\mathbf{s}$), and H_1, w_{target} and w_{test} belong to different speakers (different $\mathbf{s}$). This score can be expressed as

$$\begin{aligned} S(w_{target}, w_{test}) &= \log \frac{P(w_{target}, w_{test}|H_0)}{P(w_{target}|H_1)P(w_{test}|H_1)} \\ &= \log \frac{\int P(w_{target}, w_{test}|\mathbf{s})P(\mathbf{s})d\mathbf{s}}{\int P(w_{target}|\mathbf{s}_1)P(\mathbf{s}_1)d\mathbf{s}_1 \int P(w_{test}|\mathbf{s}_2)P(\mathbf{s}_2)d\mathbf{s}_2} \end{aligned} \tag{3.23}$$

Each item in the denominator can be rewritten as

$$\int P(w|\mathbf{s})P(\mathbf{s})d\mathbf{s} = \int P(w|\mathbf{s}, \mathbf{u})P(\mathbf{u})d\mathbf{u}P(\mathbf{s})d\mathbf{s} \tag{3.24}$$

The numerator can be rewritten as

$$\int P(w_i, w_j|\mathbf{s})P(\mathbf{s})d\mathbf{s} = \int \left[\int P(w_i|\mathbf{s}, \mathbf{u}_i)P(\mathbf{u}_i)d\mathbf{u}_i \int P(w_i|\mathbf{s}, \mathbf{u}_j)P(\mathbf{u}_j)d\mathbf{u}_j \right] P(\mathbf{s})d\mathbf{s} \tag{3.25}$$

Note that all the conditional probabilities in Equations 3.24 and 3.25 are defined in Equations 3.7, 3.8, and 3.9. Thus the log ratio score in Equation 3.23 can be easily calculated. In fact we decompose the likelihood by writing the joint likelihood of all observed and hidden variables, and then marginalize over the unknown hidden variables.

■ 3.5 Chapter Summary

In this chapter, we explained the factor analysis based speaker verification. We first introduced Joint Factor Analysis (JFA) that can correlate or link together the different Gaussian components of the UBM. Based on JFA, a simplified solution, called total variability, is presented to give the i-vector representation. Cosine similarity scoring and probabilistic linear discriminant analysis for scoring i-vectors were also introduced.

Chapter 4

Compensation Techniques

There are many variabilities among different trials, e.g., speaker identity, transmission channel, utterance length, speaking style, etc. It has been shown that these variations have a negative impact on the system performance [15]. Thus compensation techniques are needed to cope with speech variability. Successful compensation techniques have been proposed at different levels, e.g., at the feature, model, session, or score level [19] [24].

In this chapter, we will talk about the compensation techniques at the session and score level. Three approaches for session compensation are introduced. Score normalization is explained as the compensation technique at the score level. We present the motivation of score normalization and the formulations of three score normalization methods.

■ 4.1 Session Compensation

In the i-vector representation, there is no explicit compensation for inter-session variability. But the low-dimensional representation rewards compensation techniques in the new space, with the benefit of less expensive computation as well.

■ 4.1.1 Linear Discriminant Analysis (LDA)

LDA attempts to define new axes that minimize the within-class variance caused by session/channel effects, and to maximize the variance between classes. The LDA optimization problem can be defined to find direction q that maximizes the Fisher criteria

$$J(q) = \frac{\parallel q^t S_b q \parallel}{\parallel q^t S_w q \parallel} \tag{4.1}$$

where S_b and S_w are between-class and within-class covariance matrices:

$$S_b = \sum_{r=1}^{R} (\overline{w^r} - \overline{w})(\overline{w^r} - \overline{w})^t \tag{4.2}$$

$$S_w = \sum_{r=1}^{R} \frac{1}{n_r} \sum_{i=1}^{n_r} (w_i^r - \overline{w^r})(w_i^r - \overline{w^r})^t \tag{4.3}$$

and $\overline{w^r} = (1/n_r) \sum_{i=1}^{n_r} w_i^r$ is the mean of the i-vectors for each speaker, n_r is the number of utterances for each speaker r, $\overline{w}$ is the speaker population mean vector (the mean of all the available i-vectors for training), R is the number of speakers. The projection matrix A is achieved by maximizing the Fisher criteria. It is composed of the top eigenvectors of the general matrix $S_w^{-1} S_b$ [13].

The new cosine kernel between two i-vectors w_1 and w_2 can be rewritten as

$$k(w_1, w_2) = \frac{(Aw_1)^t (Aw_2)}{\sqrt{(Aw_1)^t (Aw_1)} \sqrt{(Aw_2)^t (Aw_2)}} \tag{4.4}$$

■ 4.1.2 Within-Class Covariance Normalization (WCCN)

WCCN is used as a channel compensation technique to scale a subspace to attenuate dimensions of high within-class variance [14]. It is a linear feature projection which aims to minimize the risk of misclassification of SVM classifiers [16]. The projection matrix B is obtained such that $BB^t = W^{-1}$, where W is the average of the within-class covariance matrix of all the impostors

$$W = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{n_r} \sum_{i=1}^{n_r} (w_i^r - \overline{w^r})(w_i^r - \overline{w^r})^t \tag{4.5}$$

In Equation 4.5, $\overline{w^r} = (1/n_r) \sum_{i=1}^{n_r} w_i^r$ is the mean of the i-vectors for each speaker, n_r is the number of utterances for each speaker r, $\overline{w}$ is the speaker population mean vector, R is the number of speakers. [14] has provided detailed proofs and analysis for deriving the WCCN projection.

The cosine kernel based on the WCCN matrix is given as follows

$$k(w_1, w_2) = \frac{(Bw_1)^t (Bw_2)}{\sqrt{(Bw_1)^t (Bw_1)} \sqrt{(Bw_2)^t (Bw_2)}} \tag{4.6}$$

■ 4.1.3 Nuisance Attribute Projection (NAP)

NAP is a technique to modify the kernel distance between two feature vectors via the removal of subspaces that cause undesired kernel variability [16]. The projection matrix is formulated as

$$P = I - RR^t \tag{4.7}$$

where R is a low-rank rectangular matrix whose columns are the k eigenvectors having the largest eigenvalues of the within-class covariance matrix [13].

The new cosine kernel can be rewritten as

$$k(w_1, w_2) = \frac{(Pw_1)^t(Pw_2)}{\sqrt{(Pw_1)^t(Pw_1)}\sqrt{(Pw_2)^t(Pw_2)}} \tag{4.8}$$

■ 4.2 Score Normalization

Variability compensation at the score level is also referred to as score normalization. These techniques are defined as a transformation to the output scores of a speaker verification system in order to reduce misalignments in the score ranges due to variations in the conditions of a trial. Score normalization is introduced to make a speaker-independent decision threshold more robust and effective.

The decision-making process used in speaker verification based on GMM-UBMs compares the likelihood ratio obtained from the claimed speaker model and the UBM model with a decision threshold. Due to score variability between verification trials, the choice of decision threshold is an important, and troublesome problem. Score variability mainly consists of two different sources. One is the different quality of speaker modeling caused by variation in enrollment data. Another is the possible mismatches and environment changes among test utterances.

Researchers use z-norm and t-norm to obtain a calibrated score [21]. We assume the length-normalized target speaker i-vector is w'_{target} and the length-normalized test i-vector is w'_{test}.

Z-norm calculates the scores of the target speaker model against a set of impostor speech utterances. The mean μ_{znorm} and standard deviation σ_{znorm} of these scores are estimated to normalize the target speaker score. Each target speaker has an associated μ_{znorm} and σ_{znorm}. The z-normalized score is

$$score_{znorm}(w'_{target}, w'_{test}) = \frac{score(w'_{target}, w'_{test}) - \mu_{znorm}}{\sigma_{znorm}} \tag{4.9}$$

In cosine similarity scoring, $\mu_{znorm} = {w'_{target}}^t \cdot \overline{w'}$, and $\sigma_{znorm} = \sqrt{{w'_{target}}^t \cdot C \cdot w'_{target}}$ [12]. where $\overline{w'}$ is the mean of "impostor" i-vectors, C is the impostor's covariance matrix, $C = E[(w' - \overline{w'})(w' - \overline{w'})^t]$. Thus the z-normalized score can be rewritten as

$$\begin{aligned} score_{znorm}(w'_{target}, w'_{test}) &= \frac{{w'_{target}}^t \cdot w'_{test} - {w'_{target}}^t \cdot \overline{w'}}{\sqrt{{w'_{target}}^t \cdot C \cdot w'_{target}}} \\ &= \frac{{w'_{target}}^t \cdot (w'_{test} - \overline{w'})}{\sqrt{{w'_{target}}^t \cdot C \cdot w'_{target}}} \end{aligned} \tag{4.10}$$

Similarly, t-norm parameters are estimated from scores of each test segment against a set of impostor speaker models. The mean μ_{tnorm} and standard deviation σ_{tnorm} of these scores are used to adjust the target speaker score. Each impostor speaker model has an associated μ_{tnorm} and σ_{tnorm}. The t-normalized score is

$$score_{tnorm}(w'_{target}, w'_{test}) = \frac{score(w'_{target}, w'_{test}) - \mu_{tnorm}}{\sigma_{tnorm}} \tag{4.11}$$

In cosine similarity scoring, $\mu_{tnorm} = {w'_{test}}^t \cdot \overline{w'}$, and $\sigma_{tnorm} = \sqrt{{w'_{test}}^t \cdot C \cdot w'_{test}}$. Thus the t-normalized score can be rewritten as

$$score_{tnorm}(w'_{target}, w'_{test}) = \frac{(w'_{target} - \overline{w'})^t \cdot w'_{test}}{\sqrt{{w'_{test}}^t \cdot C \cdot w'_{test}}} \tag{4.12}$$

In [12], Dehak proposed a new cosine similarity scoring. This new scoring is given as below:

$$score(w'_{target}, w'_{test}) = \frac{(w'_{target} - \overline{w'})^t(w'_{test} - \overline{w'})}{\sqrt{{w'_{target}}^t \cdot C \cdot w'_{target}}\sqrt{{w'_{test}}^t \cdot C \cdot w'_{test}}} \tag{4.13}$$

It can be treated as the combination of z-norm and t-norm score normalization, since it captures both the variabilities of different speaker models and the mismatches among different test utterances. This normalization is referred to as "combined norm" in the following chapters.

■ 4.3 Chapter Summary

In this chapter, we introduced three techniques for session compensation. These intersession compensation methods can remove the session variabilities between different trials. Two score normalization methods, t-norm and z-norm, are introduced, along with the corresponding representations in cosine similarity scoring. The new cosine similar scoring proposed by Dehak [12] is also introduced and will be used in the following experiments.

Chapter 5

Distance Metric Learning

With i-vectors as low-dimensional representations of speech utterances, a cosine distance classifier measures the distance between the target user utterance and the test utterance. Although Cosine Similarity Scoring has proven to be effective in speaker verification, we would like to explore the hidden structure of the i-vector space. Defining the distance metric between vectors in a feature space is a crucial problem in machine learning [38]. A learned metric can significantly improve the performance in classification, clustering and retrieval tasks [32] [33]. The objective of distance metric learning is to learn a distance metric that preserves the distance relation among the training data from a given collection of pairs of similar/dissimilar points [32]. Since the basic speaker verification task is to determine whether the test utterance and the target utterance are from the same speaker, a good distance metric can differentiate utterances from different speakers well, and thus achieve good performance in speaker verification.

In this chapter, we explore two supervised distance metric learning methods. As a classical distance metric learning algorithm, Neighborhood Component Analysis (NCA) is first introduced. However, the point estimation of the distance metric and the unreliability with limited training examples make NCA not as powerful as expected. Thus the Bayesian framework is presented to estimate a posterior distribution for the distance metric, which has no requirement on the number of training examples.

■ 5.1 Neighborhood Component Analysis

Neighborhood Component Analysis (NCA) [34] learns a distance metric to minimize the average leave-one-out (LOO) K-nearest-neighbor (KNN) classification error under a stochastic selection rule. The k nearest neighbor classifier identifies the labeled data points that are closest to a given test data point, which involves the estimation of a distance metric. Appropriately designed distance metrics can significantly benefit KNN classification accuracy compared to the standard Euclidean distance. We briefly review the key idea of NCA below.

Given a labeled data set consisting of i-vectors $w_1, w_2, ..., w_n$ and corresponding speaker labels $y_1, y_2, ..., y_n$, we want to find a distance metric that maximizes the performance of nearest neighbor classification. Ideally, we would like to optimize the performance on future test data, but since we do not know the true data distribution, we instead attempt to optimize the leave-one-out (LOO) performance on the training data. In what follows, we restrict ourselves to learning Mahalanobis (quadratic) distance metrics, which can always be represented by symmetric positive semi-definite matrices. We estimate such metrics through their inverse square roots, by learning a linear transformation of the input space such that KNN performs well in the transformed space. If we denote the transformation by a matrix B, we are effectively learning a metric $Q = B^T B$ such that $d(w_i, w_j) = (w_i - w_j)^t Q(w_i - w_j) = (Bw_i - Bw_j)^t(Bw_i - Bw_j)$.

The actual LOO classification error of KNN is a discontinuous function of the transformation B, since an infinitesimal change in B may change the neighbor graph and thus affect LOO classification performance by a large amount. Instead, we adopt a better behaved measure of nearest neighbor performance, by introducing a differentiable cost function based on stochastic (soft) neighbor assignments in the transformed space. In particular, each utterance w_i selects another utterance w_j as its neighbor with some probability p_{ij}, and inherits its speaker label from the utterance it selects. We define p_{ij} using a softmax over Euclidean distances in the transformed space:

$$p_{ij} = \frac{\exp(- \parallel Bw_i - Bw_j \parallel^2)}{\sum_{k \neq i} \exp(- \parallel Bw_i - Bw_k \parallel^2)}, \, p_{ii} = 0 \tag{5.1}$$

The probability for the utterance w_i selecting neighbors from the same speaker is $p_i = \sum_{j \in C_i} p_{ij}$, where C_i is the set of utterances from the same speaker with i. The projection matrix B maximizes the expected number of utterances selecting neighbors from the same speaker:

$$B = \text{argmax}_B f(B) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i \tag{5.2}$$

A conjugate gradient method is used to obtain the optimal B. Differentiating f with respect to the projection matrix B generates the gradient as below:

$$\begin{aligned}\frac{\partial f}{\partial B} &= -2B\sum_{i}\sum_{j\in C_i} p_{ij}(w_{ij}w_{ij}^T - \sum_{k} p_{ij}x_{ik}x_{ik}^T) \\ &= 2B\sum_{i}(p_i\sum_{k} p_{ik}w_{ik}w_{ik}^T - \sum_{j\in C_i} p_{ij}w_{ij}w_{ij}^T)\end{aligned} \tag{5.3}$$

■ 5.2 Bayesian Distance Metric Learning Framework

NCA provides a point estimation of the distance metric and can be unreliable when the number of training examples is small. The work in [33] presents a Bayesian framework to estimate a posterior distribution for the distance metric by applying a prior distribution on the distance metric.

Given the speaker-label of each utterance, we can form two sets of same-speaker and different-speaker constraints S and D. The probability of two utterances w_i and w_j belonging to the same speaker or different speakers is defined under a given distance matrix A:

$$P(y_{ij}|w_i, w_j, A, \alpha) = \frac{1}{1 + \exp\left(y_{ij}(||w_i - w_j||_A^2 - \alpha)\right)} \tag{5.4}$$

where
$$y_{i,j} = \begin{cases} +1 & (w_i, w_j) \in S \\ -1 & (w_i, w_j) \in D \end{cases}$$

The parameter α is the threshold used to differentiate same-speaker utterances and different-speaker utterances. Two utterances are more likely to be identified from the same speaker only when their distance with respect to the distance matrix A is less than α. The complete likelihood function for all the constraints in S and D is

$$P(S, D|A, \alpha) = \prod_{(i,j)\in S} \frac{1}{1 + \exp(\parallel w_i - w_j \parallel_A^2 - \alpha)} \times \prod_{(i,j)\in D} \frac{1}{1 + \exp(- \parallel w_i - w_j \parallel_A^2 + \alpha)} \tag{5.5}$$

We introduce a Wishart prior for the distance metric A and a Gamma prior for the threshold α as

$$P(A) = \frac{|A|^{(v-m-1)/2}}{Z_V(W)} \exp\left(-\frac{1}{2}tr(W^{-1}A)\right) \tag{5.6}$$

$$P(\alpha) = \frac{b-1}{Z(b)} \exp(-\beta\alpha) \tag{5.7}$$

where $Z_v(W)$ and $Z(b)$ are the normalization factors. Plugging the priors into the likelihood function, we can obtain the posterior distribution as follows

$$P(A, \alpha|S, D) = \frac{P(A)P(\alpha)P(S, D|A, \alpha)}{\int_A P(A)dA \int_0^\infty P(\alpha)P(S, D|A, \alpha)d\alpha} \tag{5.8}$$

The optimal A and α are obtained to maximize the posterior distribution above. But the integration over the space of positive semi-definitive matrices makes the estimation computationally intractable. Thus an efficient algorithm is necessary to compute $P(A, \alpha|S, D)$.

To simplify the computation, the distance metric A is modeled as a parametric form of the top eigenvectors of the observed data points [33]. Let $X = (w_1, w_2, \ldots, w_n)$ denote all the available utterances, and $\mathbf{v}_l, l = 1, \ldots, K$ be the top K eigenvectors of XX^T. If we assume $A = \sum_{l=1}^K \gamma_l \mathbf{v}_l \mathbf{v}_l^T$, where $\gamma_l \geq 0, l = 1, 2, \ldots, K$, the likelihood $P(y_{i,j}|w_i, w_j)$ can be rewritten as:

$$\begin{aligned} P(y_{ij}|w_i, w_j, A, \alpha) &= \frac{1}{1 + \exp\left(y_{ij}(\sum_{l=1}^K \gamma_l w_{i,j}^l - \alpha)\right)} \\ &= \sigma(-y_{i,j}\gamma^t w_{i,j}) \end{aligned} \tag{5.9}$$

where

$$w_{i,j}^l = [(w_i - w_j)^t \mathbf{v}_l]^2$$

$$w_{i,j} = (-1, w_{i,j}^1, \ldots, w_{i,j}^K)$$

$$\gamma = (\alpha, \gamma_1, \ldots, \gamma_K)$$

$$\sigma(z) = 1/(1 + \exp(-z))$$

Reducing the Wishart and gamma prior in Equation 5.6 and 5.7 into a set of Gaussian distributions on the parameters $\gamma = (\alpha, \gamma_1, \ldots, \gamma_K)$, the prior distribution is expressed as

$$\begin{aligned} P(A)P(\alpha) &= \prod_{i=1}^{K+1} N(\gamma_i; \gamma_0, \delta^{-1}) \\ &= N(\gamma; \gamma_0 \mathbf{1}_{K+1}, \delta^{-1}\mathbf{1}_{K+1}) \end{aligned} \tag{5.10}$$

Thus, the evidence function is computed as:

$$\begin{aligned} P(S, D) &= \int P(S, D|\gamma)P(\gamma)d\gamma \\ &= \int \prod_{(i,j)\in S} \sigma(-\gamma^t w_{i,j}) \prod_{(i,j)\in D} \sigma(\gamma^t w_{i,j}) N(\gamma; \gamma_0 \mathbf{1}_{K+1}, \delta^{-1}\mathbf{I}_{K+1})d\gamma \end{aligned} \tag{5.11}$$

One problem with the relaxation of the priors is that the combination weights γ are no longer guaranteed to be non-negative. But this problem is solved empirically by enforcing the mean of the γ to be non-negative.

■ 5.3 Variational Approximation

The transformation of the likelihood to a logistic function makes it possible to get a lower bound of the evidence, thus a variational method [33] [37] is employed to estimate the posterior distribution for γ. The key idea is to introduce variational distributions for γs to construct the lower bound for the logarithm of the evidence function. The approximate estimation for the posterior distribution of γs is obtained by maximizing the variational distributions with respect to the lower bound. Given the variational distribution $\phi(\gamma)$, the logarithm of the evidence function is lower bounded by the following expression

$$\begin{aligned} \log P(S, D) &= \log \int d\gamma \, P(\gamma) \prod_{(i,j)\in S} P(+|w_i, w_j) \prod_{(i,j)\in D} P(-|w_i, w_j) \\ &\geq \langle \log P(\gamma) \rangle + H(\phi(\gamma)) + \sum_{(i,j)\in S} \langle \log P(+|w_i, w_j) \rangle + \sum_{(i,j)\in D} \langle \log P(-|w_i, w_j) \rangle \end{aligned} \tag{5.12}$$

where $\langle\cdot\rangle = \langle\cdot\rangle_{\phi_\gamma}$.

Using the inequality $\sigma(z) \geq \sigma(\xi)\exp\left(\frac{z-\xi}{2} - \lambda(\xi)(z^2-\xi^2)\right)$ where $\lambda(\xi) = \frac{\tanh(\frac{\xi}{2})}{4\xi}$, we can lower bound $\langle\log P(y|w_i, w_j)\rangle$ by the following expression

$$\langle\log P(y|w_i,w_j)\rangle \geq \log\sigma(\xi_{i,j}) + \frac{-y\left\langle\gamma\right\rangle^T w_{i,j} - \xi_{i,j}}{2} - \lambda(\xi_{i,j})\left(\mathrm{tr}\left(w_{i,j}w_{i,j}^T\left\langle\gamma\gamma^T\right\rangle\right) - \xi_{i,j}^2\right) \tag{5.13}$$

Now we obtain a new expression for the lower bound of the evidence function

$$\begin{aligned} \log P(S,D) \geq\ & \langle\log P(\gamma)\rangle + H(\phi(\gamma)) \\ & + \sum_{(i,j)\in S}\left(\log\sigma(\xi_{i,j}^s) - \frac{\left\langle\gamma\right\rangle^T w_{i,j}^s + \xi_{i,j}^s}{2}\right) \\ & + \sum_{(i,j)\in D}\left(\log\sigma(\xi_{i,j}^d) + \frac{\left\langle\gamma\right\rangle^T w_{i,j}^d - \xi_{i,j}^d}{2}\right) \\ & - \sum_{(i,j)\in S}\lambda(\xi_{i,j}^s)\left(\mathrm{tr}(w_{i,j}^s[w_{i,j}^s]^T\left\langle\gamma\gamma^T\right\rangle) - [\xi_{i,j}^s]^2\right) \\ & - \sum_{(i,j)\in D}\lambda(\xi_{i,j}^d)\left(\mathrm{tr}(w_{i,j}^d[w_{i,j}^d]^T\left\langle\gamma\gamma^T\right\rangle) - [\xi_{i,j}^d]^2\right) \end{aligned} \tag{5.14}$$

Variational parameters $\xi_{i,j}^s$ and $\xi_{i,j}^d$ are introduced for every pairwise constraint in S and D, respectively. By maximizing the posterior distribution $\phi(\gamma)$ with respect to the lower bound of the evidence function, we have $\phi(\gamma) \sim N(\gamma; \mu\gamma, \Sigma_\gamma)$, where the mean μ_γ and the covariance matrix Σ_γ are computed by the following updated equations

$$\mu_\gamma = \Sigma_\gamma\left(\delta\gamma_0 - \sum_{(i,j)\in S}\frac{w_{i,j}^s}{2} + \sum_{(i,j)\in D}\frac{w_{i,j}^d}{2}\right) \tag{5.15}$$

$$\Sigma_\gamma = (\delta\mathbf{I}_K + 2\Sigma_S + 2\Sigma_D)^{-1} \tag{5.16}$$

In the above, Σ_S and Σ_D are defined as follows

$$\Sigma_S = \sum_{(i,j)\in S}\lambda(\xi_{i,j}^s)w_{i,j}^s[w_{i,j}^s]^T \tag{5.17}$$

$$\Sigma_D = \sum_{(i,j)\in D} \lambda(\xi^d_{i,j}) w^d_{i,j} [w^d_{i,j}]^T \tag{5.18}$$

The variational parameters are estimated as follows

$$\xi^s_{i,j} = \sqrt{[\mu_\gamma^T w^s_{i,j}]^2 + [w^s_{i,j}]^T \Sigma_\gamma w^s_{i,j}} \tag{5.19}$$

$$\xi^d_{i,j} = \sqrt{[\mu_\gamma^T w^d_{i,j}]^2 + [w^d_{i,j}]^T \Sigma_\gamma w^d_{i,j}} \tag{5.20}$$

Finally we conclude the EM-like iterations to update the combination weights γs:

- E-step: Given the values for the variational parameters $\xi^s_{i,j}$ and $\xi^d_{i,j}$, compute the mean μ_γ and the covariance matrix Σ_γ using Equations 5.15 and 5.16.
- M-step: Recompute the optimal value for $\xi^s_{i,j}$ and $\xi^d_{i,j}$ using Equations 5.19 and 5.20 based on the estimated mean μ_γ and covariance matrix Σ_γ.

After getting the posterior distribution $\phi(\gamma) \sim N(\gamma; \mu_\gamma, \Sigma_\gamma)$, the key question is how to compute the conditional probability $P(\pm|w_i, w_j)$. Incorporating the full distribution of γ, we can express $P(\pm|w_i, w_j)$ as

$$\begin{aligned} P(\pm|w_i, w_j) = &= \int \frac{N(\gamma; \mu_\gamma, \Sigma_\gamma)}{1 + \exp(\pm\gamma^T w_{i,j})} d\gamma \\ &\propto \int \exp(-l^{\pm}_{i,j}(\gamma)) d\gamma \end{aligned} \tag{5.21}$$

where $l^{\pm}_{i,j}(\gamma) = \log(1 + \exp(\pm\gamma^T w_{i,j})) + \frac{1}{2}(\gamma - \mu_\gamma)^T \Sigma_\gamma^{-1} (\gamma - \mu_\gamma)$. The above computation involves an integration requiring significant computation. Thus we employ the Laplacian approximation to calculate it effectively.

We first approximate the optimal solution $l^{\pm}_{i,j}(\gamma)$ by its Taylor expansion around the optimal point μ_γ, and then compute the integral using the approximated $l^{\pm}_{i,j}(\gamma)$. Since this involves solving the optimization $\gamma^{\pm}_{i,j} = \arg\min_{\gamma \geq 0} l^{\pm}_{i,j}(\gamma)$ for each data pair, which is computationally expensive when the number of data pairs is large, we further approximate the optimal solution $\gamma_{i,j}^{\pm}$ by expanding $l_{i,j}^{\pm}(\gamma)$ in the neighborhood of μ_γ as follows

$$\begin{aligned} l_{i,j}^{\pm}(\gamma) &\approx \log(1+\exp(\pm\mu_\gamma^T w_{i,j})) \pm p_{i,j}^{\pm}(\gamma-\mu_\gamma)^T w_{i,j} + \frac{1}{2}(\gamma-\mu_\gamma)^T(\Sigma_\gamma^{-1} + q_{i,j}^{\pm} w_{i,j} w_{i,j}^T)(\gamma-\mu_\gamma) \\ &\approx \log(1+\exp(\pm\mu_\gamma^T w_{i,j})) \pm p_{i,j}^{\pm}(\gamma-\mu_\gamma)^T w_{i,j} + \frac{1}{2}(\gamma-\mu_\gamma)^T \Sigma_\gamma^{-1}(\gamma-\mu_\gamma) \end{aligned} \tag{5.22}$$

where

$$p_{i,j}^{\pm} = \frac{\exp(\pm\mu_\gamma^T w_{i,j})}{1+\exp(\pm\mu_\gamma^T w_{i,j})} \tag{5.23}$$

$$q_{i,j}^{\pm} = p_{i,j}^{\pm}(1-p_{i,j}^{\pm}) \tag{5.24}$$

In Equation 5.22, $(\Sigma_\gamma^{-1}+q_{i,j}^{\pm} w_{i,j} w_{i,j}^T)$ is approximated as Σ_γ^{-1} because Σ_γ^{-1} is a summation across all the labeled example pairs according to Equation 5.16 and therefore is significantly more important than the single item $q_{i,j}^{\pm} w_{i,j} w_{i,j}^T$. Thus the approximate solutions for $\gamma_{i,j}^{\pm}$ and $l_{i,j}^{\pm}(\gamma)$ are

$$\gamma_{i,j}^{\pm} \approx \max\left(\mu_\gamma\right) \mp p_{i,j}^{\pm}\Sigma_\gamma w_{i,j}, \mathbf{0}\right) \tag{5.25}$$

$$l_{i,j}^{\pm}(\gamma) \approx l_{i,j}^{\pm}(\gamma_{i,j}^{\pm}) + \frac{(\gamma-\gamma_{i,j}^{\pm})^T \Sigma_\gamma^{-1}(\gamma-\gamma_{i,j}^T)}{2} \tag{5.26}$$

The max operator in Equation 5.25 refers to element wise maximization.

With the above approximations, the posterior $P(\pm|w_i,w_j)$ is computed as

$$P(\pm|w_i,w_j) \propto \exp(-l_{i,j}^{\pm}(\gamma_{i,j}^{\pm})) = \frac{1}{1+\exp(\pm w_{i,j}^T \gamma_{i,j}^{\pm})} \exp\left(-\frac{[p_{i,j}^{\pm}]^2 w_{i,j}^T \Sigma_\gamma w_{i,j}}{2}\right) \tag{5.27}$$

In Equation 5.27, both the mean and the covariance matrix of the distribution of γ are taken into account in the estimation of the posterior distribution. Lastly $P(\pm|w_i,w_j)$ are normalized to ensure $P(+|w_i,w_j) + P(-|w_i,w_j) = 1$. The probability of identifying the target and test utterance from the same speaker $P(+|w_{target},w_{test})$ is the output score of this approach.

■ 5.4 Chapter Summary

In this chapter, we described the Bayesian distance metric learning framework. A classical distance metric learning algorithm, Neighborhood Component Analysis (NCA), is first introduced. Since NCA can only model the distance between data points in the Euclidean space, it is unable to characterize the data points lying in a complicated space. Different from the point estimation in NCA, we aim to obtain a posterior distribution for the distance metric. The calculation of the posterior distribution involves the integration over the space of semi-definitive matrices, which is computationally intractable. We approximate the distance metric as a parametric form of the top eigenvector of the observed data points and express the likelihood as a logistic function. Applying a set of Gaussian distributions on the parameters, we can obtain a lower bound of the evidence, thus a variational method is employed to estimate the posterior distribution of the parameters. The probability of identifying the target and test utterance from the same speaker is the output score of this approach.

Chapter 6

Experimental Results

This chapter will present experimental results on the female part of the NIST 2008 SRE dataset. The parameter selection is performed first to obtain the best reduced dimension in LDA. Then the comparison between cosine similarity scoring and Bayesian distance metric learning is presented with a detailed analysis. Finally, the results with limited training data, and with short-duration data are introduced.

■ 6.1 Experimental Set-up

Experiments are performed on the female part of the NIST 2008 SRE (speaker recognition evaluation) dataset [39]. The NIST 2008 SRE released 13 different speaker detection tests defined by the duration and type of the training and test data. It includes six training conditions and four test conditions. We present results on the short2-short3 and 10sec-10sec conditions. In the short2-short3 condition (also called "core condition"), the training and test data are telephone conversational excerpts of approximately five minutes duration. In the 10sec-10sec condition, the training and test data are telephone conversational excerpts of approximately 10 seconds duration. The dataset for i-vector training contains 1,830 speakers and 21,382 utterances [40]. It is also used for LDA and NCA training, and as the impostor set in the score normalization step. A 600-dimension i-vector is extracted from each utterance. The Equal Error Rate (EER) and the minimum Detection Cost Function (minDCF) are used as metrics for evaluation.

Section 6.2, 6.3, and 6.4 describe evaluations on the short2-short3 condition, while section 6.5 describe evaluations on the 10sec-10sec condition.

■ 6.2 Parameter Selection

This section first presents the results obtained with linear discriminant analysis (LDA) applied to the i-vectors in order to compensate for channel effects. Figure 6.1 shows the results using different LDA dimensions and different score normalization techniques. From the figure, we can see that score normalization improves the minDCF significantly. The "combinednorm" performs better than both znorm and tnorm. Furthermore, the application of LDA to rotate space for minimizing the within-speaker variance improves the performance for all normalization methods. The best results are obtained by reducing the dimensionality to 200.

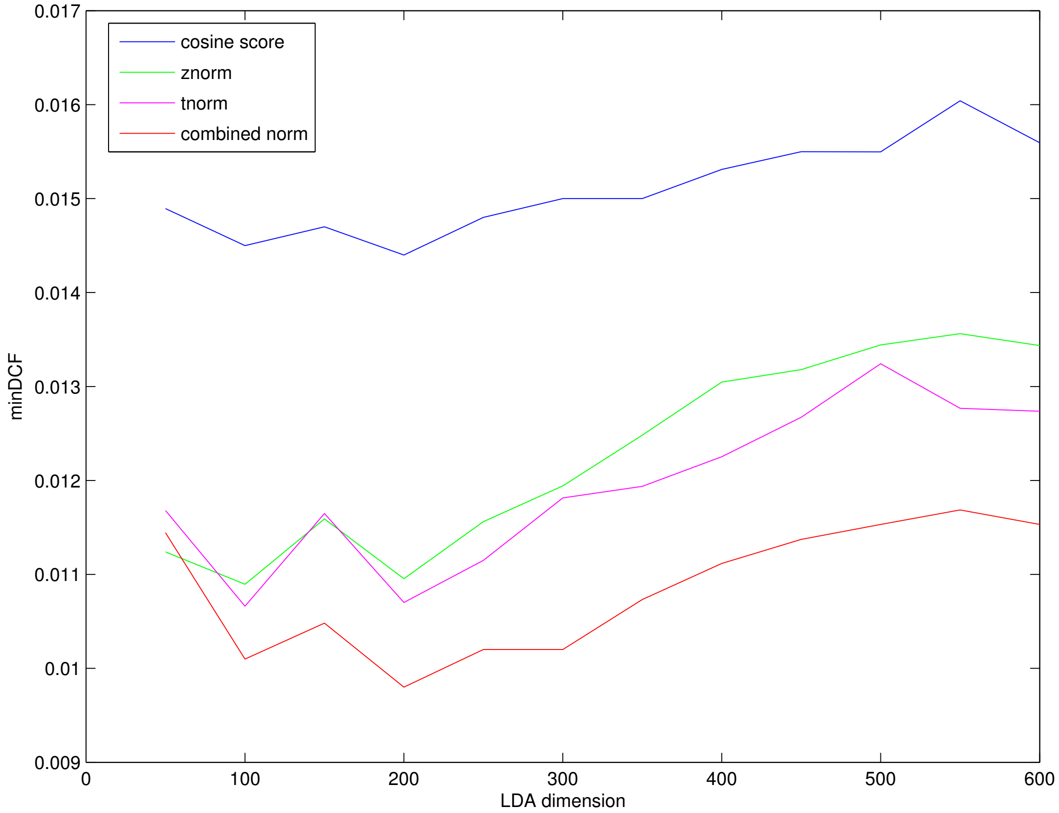

Figure 6.1: *minDCF on the female part of the core condition of the NIST 2008 SRE based on LDA technique for dimensionality reduction with different score normalization methods.*

■ 6.3 Results Comparison

In this section, we compare cosine similarity scoring and Bayesian distance metric learning on the short2-short3 condition of the NIST 2008 SRE dataset. The Bayesian distance metric learning algorithm is referred to as "Bayes_dml", cosine score after the combined score normalization described in Section 4.2 as "Cosine Score_combined norm", and PLDA with Gaussian priors as GPLDA. In Bayes_dml, we construct the similar- and different-speaker set as follows: all possible i-vector pairs from the same speaker form the constraint S; cosine scoring is applied to all possible i-vector pairs from different speakers, and those with the highest scores are selected to form the constraint D as these pairs are the most discriminative ones for a distance metric to distinguish. Since the number of all possible different-speaker pairs is extremely large, we select twice the number of similar-speaker pairs from the set of all possible different-speaker pairs to form D. Pilot experiments showed that a larger different-speaker constraint set (four or eight times the number of similar-speaker pairs) did not improve the performance but required much more computation, while a smaller different-speaker constraint set (the same size as the similar-speaker constraint set) hurt performance. The comparison is shown in Table 6.1.

Table 6.1: *Comparison of cosine score, Bayes_dml and GPLDA w/o score normalization on the female part of the core condition of the NIST 2008 SRE.*

	EER	minDCF
LDA200+Cosine Score	2.542%	0.0144
LDA200+Cosine Score_combined norm	**1.791%**	**0.0098**
LDA200+Bayes_dml	2.163%	0.0108
LDA200+Bayes_dml+znorm	2.163%	0.0108
LDA200+Bayes_dml+tnorm	2.163%	0.0108
GPLDA	3.02%	0.0157

From the table, we can see that Cosine Score_combined norm with LDA200 achieves the best result and GPLDA performs the worst. However, Bayes_dml performs better than cosine score without score normalization. Compared with the state-of-the-art performance from Cosine Score_combined norm, the gap with Bayes_dml is quite small. Furthermore, there is almost no benefit to be derived from score normalization in Bayes_dml.

The differences can be found clearly from the histograms of target scores and non-target scores from Cosine Score and Bayes_dml, which are shown in Figure 6.2 and Figure 6.3, respectively. The target scores represent the scores of test utterances from the target speaker, and the non-target scores represent the score of test utterances not from the target speaker. The score distributions from Bayes_dml are much more concentrated than those from cosine score, and the target and non-target scores are better separated as well. This comparison can explain why Bayes_dml outperforms Cosine Score in Table 6.1. As a result, there is no need to do score normalization in Bayes_dml, which makes it a more ideal model.

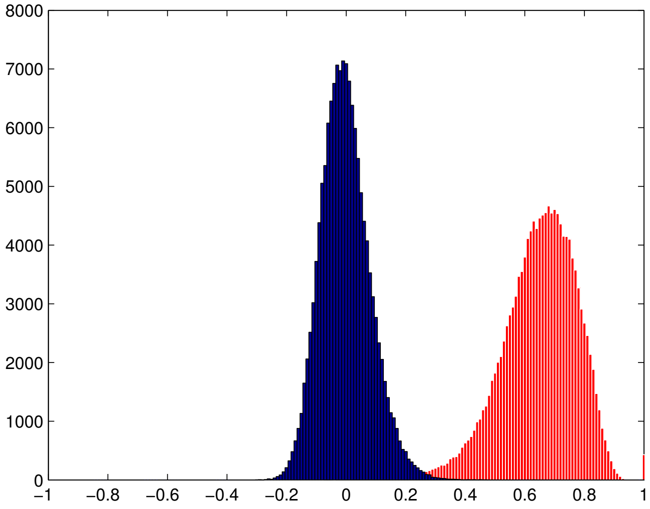

Figure 6.2: *Comparison of score histograms from Cosine Score (blue: non-target scores, red: target scores).*

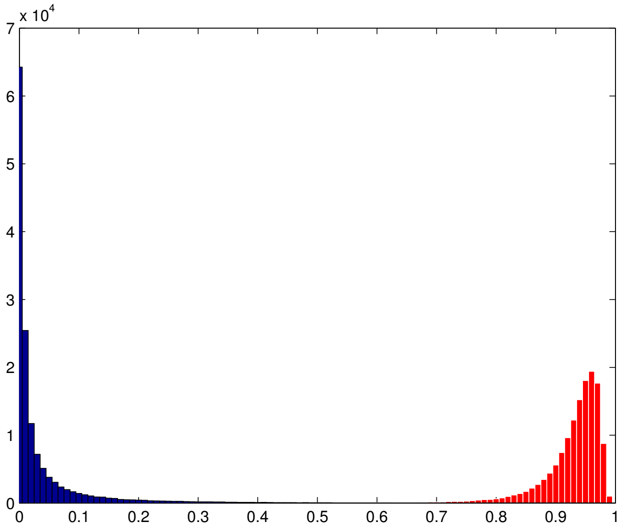

Figure 6.3: *Comparison of score histograms from Bayes_dml (blue: non-target scores, red: target scores).*

With a basic understanding of the difference between Cosine Score_combined norm and Bayes_dml, we compare their performances with different combinations of preprocessing techniques. The preprocessing techniques include LDA and NCA, which are applied before the scoring models. The results are shown in Table 6.2.

Table 6.2: *Comparison of Cosine Score_combined norm and Bayes_dml with different preprocessing techniques on the female part of the core condition of the NIST 2008 SRE.*

Cosine Score_combined norm	EER	minDCF
LDA200	1.791%	0.0098
LDA200+NCA150+LDA150	50.478%	0.1000
LDA200+NCA200	2.542%	0.0139
LDA200+NCA200+LDA100	2.018%	0.0099
LDA200+NCA200+LDA200	1.781%	0.0097
LDA600+NCA200+LDA200	42.074%	0.0099
NCA200+LDA200	4.673%	0.0287
Bayes_dml	EER	minDCF
LDA200	2.163%	0.0108
LDA200+NCA150+LDA150	41.479%	0.1000
LDA200+NCA200	3.031%	0.0178
LDA200+NCA200+LDA100	1.777%	0.0096
LDA200+NCA200+LDA200	1.815%	0.0101
LDA600+NCA200+LDA200	42.854%	0.1000
NCA200+LDA200	3.553%	0.0183

This table can give us some understanding of how NCA and LDA work in representing the hidden structure in the total variability space. The worst performance appears in the second and sixth rows.In these two cases, the dimension of NCA is different from the dimension of the previous LDA. That is to say, NCA plays a role of reducing dimensions, and it seriously affects the results. In the fourth row, NCA200 following LDA200 only makes a rotation and LDA100 afterwards reduces the dimension of feature space further, which does not hurt the performance too much. The results in the seventh row are almost in the same level with other rows except the second and fourth row, although there is a dimension reduction of NCA200 on the 600-dimension i-vectors. The reason may be that this dimension reduction is conducted in the original total variability space, while the dimension reductions in the second and fourth row are performed in the reduced feature space after LDA. The improvements in the fourth and fifth row compared to the third row show that the LDA projection corrects the feature space directions learned from NCA. Thus we can conclude that NCA does not play an effective role in dimensionality reduction.

The best performance for Cosine Score_combined norm is achieved with LDA200+NCA200+LDA200, and the best performance for Bayes_dml is achieved with LDA200+NCA200+LDA100. Bayes_dml outperforms Cosine Score_combined norm, and is also the best reported result on the short2-short3 condition of the NIST 2008 SRE female data. If we only do NCA projection, the results get worse. This is because the NCA matrix is obtained under the best nearest neighbor classification criterion without taking into consideration the clustering of i-vectors from the same speaker and the separation of i-vectors from different speakers. While LDA can achieve this goal by optimizing the Fisher criteria, generally NCA followed by LDA can project the data into a space in which i-vectors from the same speaker are closer, and i-vectors from different speakers are better separated.

Table 6.3 makes a comparison of Bayes_dml and the state-of-the-art performance. We select results presented in the literature that used the same test set, i.e. the female part of the core condition of the NIST 2008 SRE. It can be shown that Bayes_dml outperforms i-vector based SVM and GPLDA.

Table 6.3: *Comparison of results from other literatures on the female part of the core condition of the NIST 2008 SRE.*

approach	EER	minDCF
i-vector Bayes_dml+LDA200+NCA200+LDA100	1.78%	0.0096
i-vector SVM+LDA200+WCCN [13]	3.68%	0.0140
GPLDA [29]	3.13%	0.0168

■ 6.4 Results on Limited Training Data

In this section, we show the advantage of Bayes_dml when the number of training utterances for each speaker is very limited. We select three utterances from each training speaker to build a made-up training set. The test set is the same as before. The best preprocessing techniques from Section 6.3 are evaluated, with the results shown in Table 6.4.

We can see that Bayes_dml generally achieves a better EER, which means that a lower false alarm and a lower miss probability can be achieved at the same time in Bayes_dml. The best performance of Bayes_dml is better than that of Cosine Score_combined norm. Even with only 3 utterances from each speaker, we can still get rich information from same-speaker and different-speaker i-vector pairs, whereas data sparsity can cause LDA unable to fully capture the speaker variability.

Table 6.4: *Comparison of Cosine Score_combined norm and Bayes_dml on the female part of the core condition of NIST 2008 SRE with limited training data (the number of training utterances for each speaker is 3).*

Cosine Score_combined norm	EER	minDCF
LDA200	4.181%	0.0210
LDA200+NCA200+LDA200	3.930%	0.0210
LDA200+NCA200+LDA100	4.664%	0.0260
Bayes_dml	EER	minDCF
LDA200	4.514%	0.0237
LDA200+NCA200+LDA200	4.190%	0.0261
LDA200+NCA200+LDA100	3.751%	0.0208

■ 6.5 Results on Short Duration Data

Robust speaker verification on short duration utterances remains a key problem since many real applications often have access to only short duration speech data [27]. Recent studies focused on JFA have shown that performance degrades significantly in very short utterances [22] [23]. This section will present the advantage of the Bayesian distance metric learning framework for short utterances.

Table 6.5 compares Cosine_combined norm and Bayes_dml on the female part of the 10sec-10sec condition of the 2008 NIST SRE, along with some results from the literature. Bayes_dml following LDA200+NCA200+LDA100 achieves the best performance in both EER and minDCF. Although Kenny [17] has shown the superiority of heavy-tailed PLDA over Gaussian PLDA, heavy-tailed PLDA is not as effective as Bayes_dml.

Table 6.5: *Comparison of Cosine Score_combined norm and Bayes_dml on the female part of the 10sec-10sec condition of NIST 2008 SRE.*

Cosine Score_combined norm	EER	minDCF
LDA200	11.31%	0.0532
LDA200+NCA200+LDA200	10.73%	0.0534
LDA200+NCA200+LDA100	10.87%	0.0532
Bayes_dml	EER	minDCF
LDA200	10.42%	0.0567
LDA200+NCA200+LDA200	10.08%	0.0515
LDA200+NCA200+LDA100	9.955%	0.0509
GPLDA [29]	16.40%	0.0705
heavy-tailed PLDA [17]	10.9%	0.053

6.6 Chapter Summary

In this chapter, we have shown some experimental results on the female part of the NIST 2008 SRE dataset. Bayes_dml achieved comparable performance with cosine scoring, while Bayes_dml is robust to score normalization. This is because the score distributions from Bayes_dml are much more concentrated than those from cosine scoring, and the target scores and non-target score are better separated as well. Under some specific preprocessing technique, Bayes_dml outperformed cosine scoring. With limited training data and for short utterance data, Bayes_dml obtained better performance than cosine scoring. This advantage is particularly important for realistic speaker verification systems, as it can be difficult to collect plenty of samples from every speaker in many industrial applications, although possible to collect samples from a large number of different speakers.

Chapter 7

Conclusion and Future Work

■ 7.1 Summary and Contributions

In this thesis, we have proposed a Bayesian distance metric learning framework using i-vectors for speaker verification. This methodology was shown to be comparable to the state-of-the-art technique on a standard task. In Chapter 2, we described the speech parameterization to transform a speech utterance to a sequence of MFCC feature vectors for statistical modeling. We also presented the GMM-UBM approach, the classical statistical modeling approach for speaker recognition. In Chapter 3, we explained factor-analysis-based speaker verification. We introduced Joint Factor Analysis that jointly processes the different Gaussian components of the UBM. A simplified solution, called total variability, is presented that gives rise to the i-vector representation. Cosine similarity scoring and probabilistic linear discriminant analysis are used for scoring the i-vectors. Chapter 4 introduced the compensation techniques at the session and score level, since the i-vector representation contains many variable factors and there is no compensation for inter-session variability, compensation techniques are necessary to reduce the variation.

The main contributions of this thesis are detailed in Chapter 5. We proposed the Bayesian distance metric learning framework (Bayes_dml) for speaker verification. In contrast to the point estimation used in classical distance metric learning algorithms like Neighborhood Component Analysis, with Bayes_dml we aim to obtain a posterior distribution for the distance metric. The calculation of the posterior distribution involves the integration over the space of semi-definitive matrices, which is computationally intractable. We approximate the distance metric as a parametric form of the top eigenvector of the observed data points, and express the likelihood as a logistic function. Applying a set of Gaussian distributions on the parameters, we can obtain a lower bound of the evidence, thus a variational method is employed to estimate the posterior distribution of the parameters. The probability of identifying the target and test utterance from the same speaker is the output score of this approach. The experimental results detailed in Chapter 6 showed that Bayes_dml achieved comparable performance with cosine scoring, while Bayes_dml is robust to score normalization. With limited training data and for short utterance data, Bayes_dml obtained better performance than cosine scoring. These properties make Bayes_dml a very promising technique for speaker verification in real applications.

■ 7.2 Future Direction

The Bayesian distance metric learning method has shown superior performance, either in the robustness to score normalization or in short-duration utterances. We suggest several key ways in which the framework may be improved.

In the derivation of the Bayesian distance metric learning framework, we used the approximation of the distance metric rather than the direct estimation, because the calculation of the posterior distribution involves the integration over the space of semi-definitive matrices. The ultimate goal is to estimate the distance metric that can represent the characteristics of data points, thus there is no need to do channel compensation any more.

Since the cosine distance measure has very competitive performance, and distance metric learning uses Euclidean distance in the space projected by $A^{\frac{1}{2}}$, we would like to explore incorporating the cosine distance measurement into the distance metric learning framework.

The performance of speaker verification in arbitrary durations has become a critical issue in the NIST evaluation protocol since 2012. We have shown some results on short-duration utterances in this thesis, but it is still worthwhile to see how the framework works for utterances of different durations.

Bibliography

[1] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacrtaz, and D. Reynolds, "A Tutorial on Text-Independent Speaker Verification", *Eurasip Journal on Advances in Signal Processing*, vol. 2004, no. 4, pp. 430-451, 2004.

[2] D. Reynolds and R. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, 1995.

[3] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing* , vol. 10, no. 1-3, pp. 19-41, 2000.

[4] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A Study of Interspeaker Variability in Speaker Verification", *IEEE Transaction on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980-988, 2008.

[5] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and Session Variability in GMM-based Speaker Verification", *IEEE Transaction on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1448-1460, 2007.

[6] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice Modeling with Sparse Training Data", *IEEE Transaction on Speech and Audio Processing*, vol. 13, no. 3, pp. 345-354, 2005.

[7] L. Burget, N. Brummer, D. Reynolds, P. Kenny, J. Pelecanos, R. Vogt, F. Castaldo, N. Dehak, R. Dehak, O. Glembek, Z. Karam, J. N. Jr., E. Na, C. Costin, V. Hubeika, S. Kajarekar, N. Scheer, and J. Cernocky, "Robust Speaker Recognition Over Varying Channels", Johns Hopkins University, Center for Language and Speech Processing, Summer Workshop, Tech. Rep., 2008.

[8] N. Dehak, "Discriminative and Generative Approaches for Long- and Short-Term Speaker Characteristics Modeling: Application to Speaker Verification", Ph.D. Dissertation, Ecole de Technologie Superieure de Montreal, QC, Canada, June, 2009.

[9] S. Shum, "Unsupervised Methods for Speaker Diarization", S. M. Thesis, MIT Department of Electrical Engineering and Computer Science, June, 2011.

[10] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint Factor Analysis versus Eigenchannels in Speaker Recognition", *IEEE Transaction on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435-1447, 2007.

[11] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification", in *Proceedings of Interspeech*, 2009.

[12] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine Similarity Scoring without Score Normalization Techniques", in *Proceedings of IEEE Odyssey*, 2010.

[13] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788-798, 2011.

[14] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class Covariance Normalization for SVM-based Speaker Recognition", in *Proceedings of ICSLP*, 2006.

[15] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM Based Speaker Verication Using a GMM Supervector Kernel and NAP Variability Compensation", in *Proceedings of ICASSP*, 2006.

[16] H. Lei, "NAP, WCCN, a New Linear Kernel, and Keyword Weighting for the HMM Supervector Speaker Recognition System", *http://www.icsi.berkeley.edu/pubs/techreports/tr-08-006.pdf*.

[17] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors", in *Proceedings of IEEE Odyssey*, 2010.

[18] S. Prince, and J. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity", in *International Conference on Computer Vision*, 2007.

[19] P. Bousquet, D. Matrouf, and J-F. Bonastre, "Intersession Compensation and Scoring Methods in the i-vectors Space for Speaker Recognition", in *International Conference on Speech Communication and Technology*, 2011.

[20] P. Matejka, O. Glembek, F. Castaldo, M. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cemocky, "Full-covariance UBM and Heavy-tailed PLDA in i-vector Speaker Verification", in *Proceedings of ICASSP*, 2011.

[21] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification Systems", *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42-54, 2000.

[22] M. McLaren, R. Vogt, B. Baker, and S. Sridharan, "Experiments in SVM Based Speaker Verification Using Short Utterances", in *Proceedings of IEEE Odyssey*, 2010.

[23] R. Vogt, B. Bakerr, and S. Sridharan, "Factor Analysis Subspace Estimation for Speaker Verification with Short Utterances", in *Proceedings of Interspeech*, 2008.

[24] R. Vogt, S. Kajarekar, and S. Sridharan, "Discriminant NAP for SVM Speaker Recognition", in *Proceedings of IEEE Odyssey*, 2008.

[25] S. Shum, N. Dehak, R. Dehak, and J. Glass, "Unsupervised Speaker Adaptation Based on the Cosine Similarity for Text-independent Speaker Verification", in *Proceedings of IEEE Odyssey*, 2010.

[26] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, "Comparison of Scoring Methods Used in Speaker Recognition with Joint Factor Analysis", in *Proceedings of ICASSP*, 2009.

[27] J. Domnguez, R. Zazo, and J. Gonzalez-Rodrguez, "On the Use of Total Variability and Probabilistic Linear Discriminant Analysis for Speaker Verification on Short Utterances", in *Proceedings of IberSPEECH*, 2012.

[28] R. Vogt and S. Sridharan, "Explicit Modeling of Session Variability for Speaker Verification", Computer Speech Language 22(1), 1738 (2008).

[29] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "i-vector Based Speaker Recognition on Short Utterances", in *Proceedings of Interspeech*, 2011.

[30] L. Burget, P. Matejka, V. Hubeika, and J. Cernocky, "Investigation into Variants of Joint Factor Analysis for Speaker Recognition", in *Proceedings of Interspeech*, 2009.

[31] D. Rubin and D. Thayer, "EM Algorithms for ML Factor Analysis", *Psychometrika*, vol. 47, no. 1, pp. 69-76, 1982.

[32] L. Yang, "Distance Metric Learning: A Comprehensive Survey", *http://www.cs.cmu.edu/∼liuy/frame_survey_v2.pdf.*

[33] L. Yang, R. Jin, and R. Sukthankar, "Bayesian Active Distance Metric Learning", in *Uncertainty in Artificial Intelligence*, 2007.

[34] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighborhood Component Analysis", in *Neural Information Processing Systems*, 2004.

[35] R. Vogt, S. Kajarekar, and S. Sridharan, "Discriminant NAP for SVM Speaker Recognition", in *Proceedings of IEEE Odyssey*, 2008.

[36] J. Pelecanos and S. Sridharan, "Feature Warping for Robust Speaker Verification", in *Proceedings of IEEE Odyssey*, 2001.

[37] T. Jaakkola and M. Jordan, "Bayesian Parameter Estimation via Variational Methods", *Statistics and Computing*, vol. 10, no. 1, pp. 25-37, 2000.

[38] E. Xing, A. Ng, M. Jordan and S. Russell, "Distance Metric Learning with Application to Clustering with Side-Information", in *Neural Information Processing Systems*, 2002.

[39] "2008 NIST Speaker Recognition Evaluation Plan", *http://www.itl.nist.gov/iad/mig/tests/sre/2008/index.html.*

[40] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel, “An i-vector Extractor Suitable for Speaker Recognition with Both Microphone and Telephone Speech”, in *Proceedings of IEEE Odyssey*, 2010.

[41] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk, “Speech Recognition Using MFCC”, in *Proceedings of International Conference on Computer Graphics, Simulation and Modeling*, 2012.