Non-parametric Discriminate Analysis based Dimension Reduction Technology in Speaker Recognition

Author: Zhou Jianwei, S220398

Tutor: Pietro Laface

Cumani Sandro

Politecnico di Torino

November 28, 2017

Abstract

This thesis report represents firstly an overview of speaker recognition technologies, classical and new methods of automatic text-independent speaker verification namely ASV which includes a few representative techniques from 1990s until today. Additionally, recent techniques are given emphasis to have presented a paradigm shift from the traditional vector-based speaker models to so-called i-vector models.

Conventional speaker recognition system pipeline starts from feature extraction, which deals with raw speech data by extracting acoustic feature. Feature Engineering is widely concerned as the most important part of pattern recognition, speech signal includes many features of which not all are important for speaker discrimination.

Significant advancements in speaker recognition field have been made over the past years. The research trend in this area has gradually evolved from Gaussian mixture model (GMM) to joint factor analysis (JFA) based method, which attempts to model the speaker and channel spaces separately, and towards the identity vector (i-vector) approach that models both speaker and channel variability in a common single low-dimensional (e.g., often a few hundred) subspace termed the total variability subspace. Usually i-vector systems employ universal background models (UBM) to generate frame-level soft alignments required in i-vector estimation process, it extends and model the traditional super-vector. The i-vectors are typically

post-processed through linear discriminate analysis (LDA) stage to do dimension reduction and generate channel-compensated features which can then be efficiently modelled and scored with various back-end classifier such as a probabilistic LDA (PLDA).

The issue of dimension reduction of i-vector plays an important role in both the efficiency and accuracy in the processing of Probabilistic Linear Discriminant Analysis, which is an classification model in the downward of i-vector extraction. In this thesis, we report on the latest advancement in pre-processing before PLDA modelling and scoring. Particularly, a nearest-neighbour based discriminate analysis approach named NDA is introduced. NDA is used for channel compensation in i-vector space, which, different from the traditional Fisher LDA, is a non-parametric algorithm and typically of full-rank. The NDA is much more effective (up to 35% improvement in terms of EER) than the parametric LDA for speaker recognition according to the experiment.

Section 1 provides fundamentals of speaker recognition. Sections 2 and 3 then elaborate feature extraction and speaker modelling principles, as well as the current i-vector and probabilistic LDA classifiers. Section 4 and 5 describe session compensation normalization issues and dimension reduction, especially fisher LDA and NDA algorithm. Finally, experiment results between NDA and conventional LDA are outlined in Section 6, followed by conclusions in Section 7.

Contents

1 Introduction **2**

1.1 Speaker Recognition Overview 5

1.2 Selection of Features 8

1.3 Speaker Modelling 11

2 Feature Extraction **14**

2.1 Filterbank-based cepstral parameters 15

3 Speaker model and classifier **20**

3.1 GMM-UBM system 21

3.1.1 Gaussian Mixture Models 22

3.1.2 Universal Background Model 26

3.1.3 Adaptation of Speaker Model 27

3.2 Support vector machine using GMM super-vector 30

3.3 I-vector system 32

3.3.1 Total Variability 32

3.4 PLDA classifier 36

3.4.1 Gaussian PLDA (G-PLDA) 38
3.4.2 Verification score 39
3.5 Deep neural networks for extracting baum welch statistics . . 40
3.5.1 DNNs for ASR 41
3.5.2 A DNN/i-vector framework 42

4 Normalization technique **45**
4.1 Feature normalization 45
4.2 Score normalization 47
4.2.1 Znorm 48
4.2.2 Tnorm 49

5 Pre-process and dimension reduction **50**
5.1 Linear Discriminant Analysis 51
5.2 Non-parametric discriminant analysis 55

6 Implementation and experiment result **60**

7 Conclusions **64**

Chapter 1

Introduction

Being able to speak to your personal computer or mobile phone and having them recognize and understand what you say, would provide a comfortable and natural form of communication. It would also reduce the amount of typing you have to do, leave your hands free and allow you to move away from the terminal and screen. You would not even have to be in front of the terminal. Speech recognition would also help in some cases if the computer could tell *who* was speaking.

As discussed in [1] and [2], the voice signal conveys information related to the physiological characteristics of the loudspeaker as it reflects the unique characteristics such as channel, mouth, nose, size and shape. It also contains information about the behaviour aspects of a speaker such as accent, acoustical parameters like involuntary transformation. Thus, voice samples are often used as biometrics In the real world. *Speaker recognition* is the process of automatically recognizing the speakers from his / her sound sam-

ples. The speaker recognition activity can be divided into two main tasks, speaker recognition (SI) and speaker verification (SV). The speaker recognition is a pattern recognise a given set of speakers from the input speech signal. Automatic Speaker Verification (ASV) solves the identity problem of identity identity claimed in his / her voice samples. Voice-based vocabulary content, ASV systems can be widely classified as text-dependent (TD) and text-independent (TI) types. TD-ASV requires the same vocabulary content as enrolment and testing. In the case of TI-ASV, there is no restriction on the text/voice content of the voice.

Speakers identification has been most commonly used as a safety device to control access to buildings or information. One of the most famous examples is Texas Instruments' computer center security system. Secure Pacific has used the speaker to verify the security mechanism as a large amount of money transfer initiated by telephone. In addition to increasing security, validation is beneficial because it reduces the turnaround time for these bank transactions. The Bell core uses speaker verification to restrict remote access to training information to authorized field personnel. The speaker recognition also provides a mechanism to restrict the remote access of a personal workstation to its owner or a group of registered users.

Forensic analysts as well as ordinary persons benefits from speaker recognition technology. The trend goes in the direction that telephone-based services integrated speech/ speaker / language recognition will supplement or even further replace human operated telephone services in the future. Telephone conference, voice print security check, post-sale customer feedback are

some of the examples. The advantages of such automatic services are clear: much higher capacity compared to human operated services with hundreds or thousands of phone calls being processed simultaneously. In fact, the focus of speaker recognition research over the years has been tending towards such telephony-based applications.

In addition to telephone voice data, the supply of other spoken language files is increasing, such as television broadcasting, teleconferencing and video clips from holidays. Extracting meta-data from these documents, such as discussion topics or participant names and gender, will automatically perform information search and indexing. *Speaker diarisation* is also known as "who spoke when", trying to extract the accent of different participants from a spoken document, and is an extension of the classic speaker recognition technology applied to recording with multiple speakers.

In forensic and spokesperson litigation, speakers can be considered non-cooperatives because they do not particularly want to be recognized. On the other hand, in telephone-based service and access control, the user is considered to be cooperative. On the other hand, the speaker recognition system can be divided into *text-dependent* and *text-independent*. In a text-dependent system suitable for a collaborative user, the recognition phrase is fixed, or is known in advance. For example, the user may be prompted to read a randomly selected sequence of numbers, for example, in a separate system in the text, the words that allow the use of the speaker are not constrained. Thus, the reference (the content of the training) and the test (described in actual use) discourse may have completely different content,

and the recognition system must take into account this voice mismatch. Text independent recognition is more challenging for both tasks.

In general, speech variability represents an unfavourable factor in the accuracy of text-independent speaker recognition. Changes in sound environment and technical factors (sensors, channels) and "speaker" changes in his / her own (health, mood, ageing) represent other adverse factors. Often, any change between two recordings of the same speaker is called *session variability*. Conversational variability is often described as mismatched training and test conditions, and it is still the most challenging problem in speaker recognition.

1.1 Speaker Recognition Overview

Each speaker recognition system has two phases: registration and verification. During registration, the sound of the speaker is recorded, and a plurality of features are typically extracted to form a voice print, a template, or a model. During the verification phase, the voice sample or "utterance" is compared with the previously created voice print. For the recognition system, the utterance is compared with multiple voice prints in order to determine the best match, and the verification system compares the utterance to a single voice print.

Figure 1.1 is referred from [3] , which displays the components of the automatic speaker recognition system. The upper layer is the enrolment process, the lower layer shows the identification process. In the registration

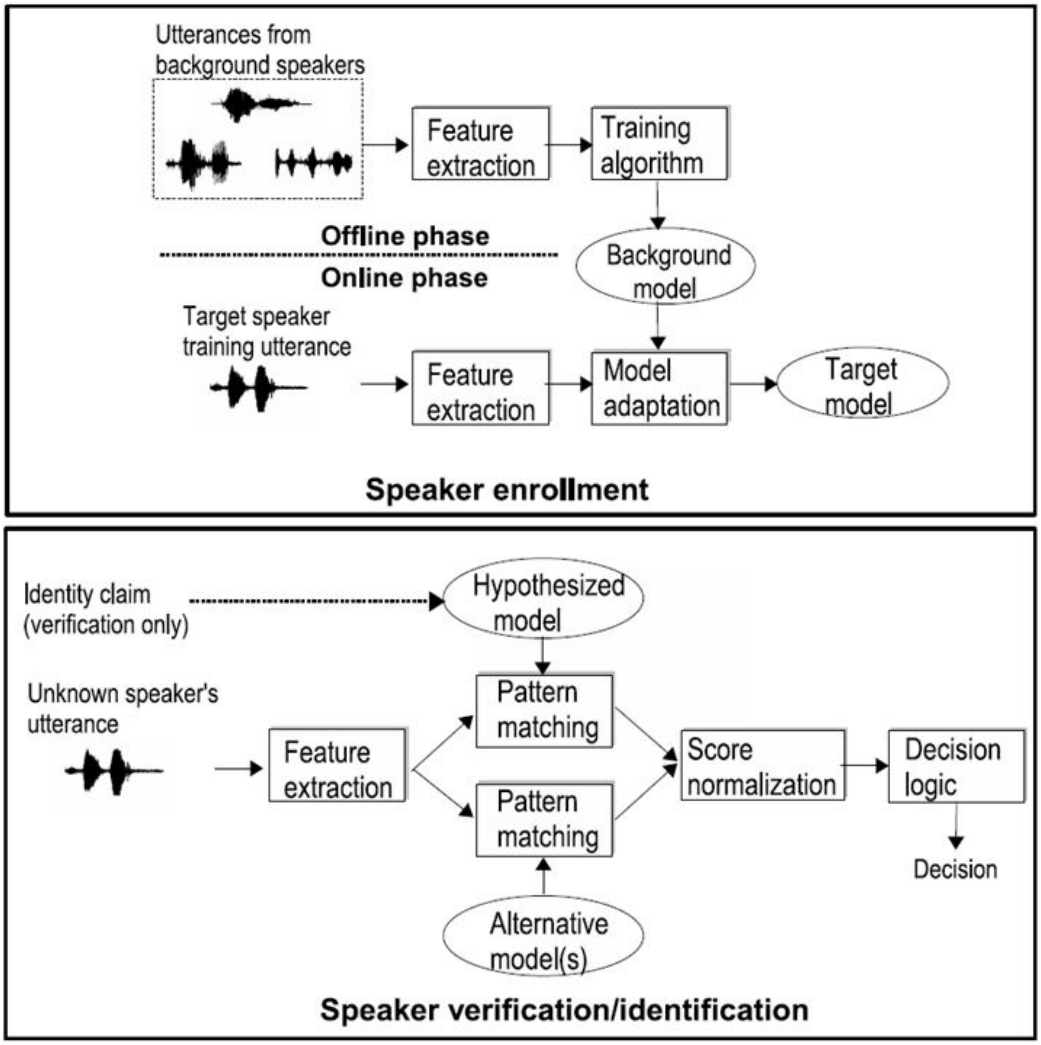

Figure 1.1: Components of a typical automatic speaker verification system

mode, a speaker model is created with the previously created background model and trained using the feature vector of the target speaker. In the recognition mode, the feature vector extracted from the utterance of the unknown person is compared with the model in the system database to obtain the similarity fraction. The decision module uses this similarity score to make the final decision.

The feature extraction module first transforms the original signal into a feature vector, which emphasizes the speaker's specific properties and suppresses the statistical redundancy. Almost all of the most advanced speaker recognition systems use a set of *background speakers* or *cohort speaker* in one or the other to enhance the robustness and computational efficiency of the recognizer. In the enrolment stage, the background speaker is used as a negative example of training discriminatory patterns, or for training the *general background model* of the target speaker model. In the identification phase, the background speaker is used to normalize the speaker's score.

The general method consists of five steps: digital voice data acquisition, feature extraction, pattern match, make an acceptance / rejection decision, and register to generate a speaker reference model. Feature extraction maps each speech interval to a multidimensional feature space. (The voice interval usually spans 10-30 ms of the speech waveform, called the speech frame). The feature vector sequence is then compared with the speaker model by pattern matching. This results in a matching score for each vector or vector sequence. The matching score measures the similarity of the calculated input feature vector to the model of the speaker or feature vector model that is

claimed to be protected by the speaker. Finally, the decision is to accept or reject the claimant in the order of the matching score or the matching score, which is a hypothetical test question.

1.2 Selection of Features

Feature extraction is the estimation of variables, called a feature vector, from another set of variables The efficient feature extraction technique extracts the feature which is able to discriminate one pattern from another accurately.

Initially, the sound sonic is transformed into a digital signal suitable for speech processing. A microphone or telephone handset can be used to convert acoustic waves into analogue signals. The analogue signal is adjusted by anti-aliasing filtering (which may require additional filtering to compensate for any channel damage). The anti-aliasing filter limits the bandwidth of the signal to approximately Nyquist rate (half of the sample rate) before sampling. The adjusted analogue signal is then sampled to form a digital signal through an analogue-to-digital (A / D) converter.

In the local speaker verification application, the analogue channel is only a microphone, its cable and analogue signal conditioning. Thus, the resulting digital signal can be of very high quality, lacking distortion caused by the transmission of analogue signals over long distance telephone lines.

Feature selection is to convert these observation vectors into feature vectors. The goal of feature selection is to find transformations of relatively low-dimensional feature spaces, to retain information about the application,

and to make meaningful comparisons using simple similarity measurements.

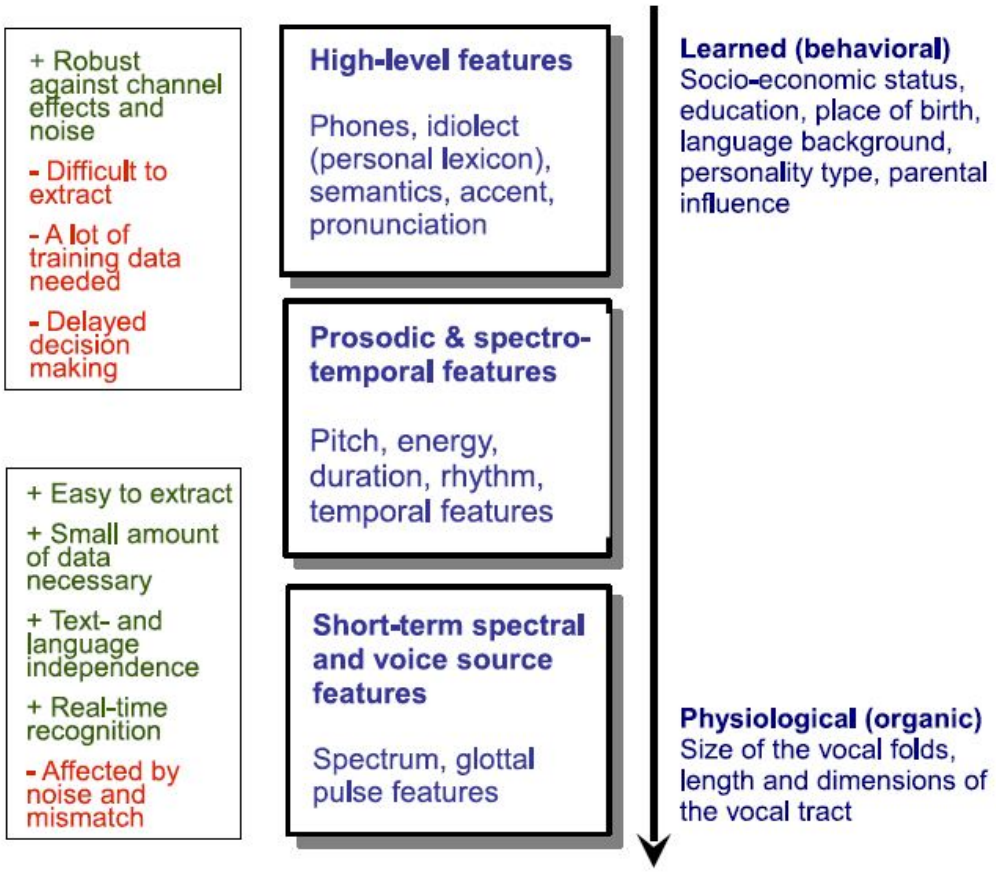

Figure 1.2: A summary of features from viewpoint of their physical interpretation.

The most advanced ASV system uses three main types of feature extraction techniques: segmentation, segmentation, and overrun analysis. The use of frame size analysis of voice signals and 3-5 ms of the mobile range is called sub-segment analysis. Studies have shown that speaker-specific excitation source information captured using sub-segment analysis contains quite

a number of speaker-specific information.

In the case of segment analysis, the speech extraction frame size indicates the channel information of the channel and the offset of the range of 10-30 ms. When the size and displacement frames are maintained in the range of 10-30 ms, it can be assumed that the speaker-specific channel information is unstable for the actual analysis and processing. The studies carried out in the segmentation function are used to extract the channel information to verify the loudspeaker.

In the extra-stage feature extraction, the frame size is used to truncate the speech and is shifted in the range of 100-300 ms. First, the technique is used to analyse and extract the characteristics of the speaker's behavioural characteristics. These combinations of information, such as word duration, intonation, spoken speed, accent, and so on. The related work shows that the super-segmentation analysis can be used to capture some behavioural characteristics and prove that it is valid for verifier verification.

The most advanced ASV system mainly uses short-term spectral characteristics. The Dayton frequency cepstrum coefficient (MFCC), perceived linear prediction (PLP) and linear predictive cepstral coefficient (LPCC) are widely used feature extraction techniques because they have considerable performance and low computational complexity.

In section 2 we would put much attention on the introduction of short-term spectral features, as they are still most widely used feature selections in modern speaker recognition system.

1.3 Speaker Modelling

By using the feature vector extracted from the training discourse of a given speaker, the speaker model is trained and stored in the system database. In the text-dependent model, the model is discourse-specific, which includes the time dependence between the eigenvectors. We often model feature distribution, the shape of the "characteristic cloud", rather than the time-dependent. In text-dependent recognition, we can align the test and training discs in time as they contain (assuming to include) the same phoneme sequence. However, in text-independent recognition, the alignment of the frame level is not possible because there is little or no correspondence between the frames in the test and reference speech. Thus, dividing the signal into a telephone or a wide range of voice categories can be used as a preprocessing step, or a speaker model can be constructed with speech.

There are two types of models: stochastic models and template models. In a stochastic model, pattern matching is probabilistic and results in a measure of the likelihood or conditional probability of a given model. For template models, pattern matching is deterministic. The training and test feature vectors are directly compared with each other, assuming that any one is another imperfect copy. In a stochastic model, each speaker is modelled as a probability source with an unknown but fixed probability density function. The training phase is a parameter that estimates the probability density function from the training sample. Matching is usually done by assessing the likelihood of a test discourse about the model.

The speaker recognition community has found a robust method of using a single vector (the so-called super vector) to present the discourse. One of the questions in speaker recognition is how to express discusser with different numbers of eigenvectors in general. In general, "super vector" refers to combining a vector of a number of smaller dimensions into a higher dimension vector; for example, the d-dimensional mean vector of the GMM adapted to the K component is stacked into a Kd-dimensional Gaussian super vector.

The speaker recognition area has made significant progress over the past few years. The research trends in this field have evolved from a method based on joint factor analysis (JFA), which attempts to model loudspeakers and channel subspaces to simulate i-vector methods that change loudspeakers and channels to unidirectional low-dimensional (for example, Hundreds of) space is called the total variability subspace. The most advanced i-based speaker recognition system uses the generic background model (UBM) to generate the required soft-pitch for the i-vector estimation process. The i-vector is usually post-processed by the linear discriminant analysis (LDA) stage to produce dimension reduction and channel compensation characteristics, which can then be effectively modelled and scored using various back ends, such as probability LDA (PLDA).

In addition, inspired by the success of the deep neural network (DNN) acoustic model in the field of automatic speech recognition (ASR), the use of DNN senone (context dependent triphones) posteriors soft alignments is proposed to significantly reduce the speaker recognition error rate.

In section 3, various speaker models will be discussed in detail, from

GMM-UBM , UBM-SVM model to i-vector-PLDA pipeline.

Chapter 2

Feature Extraction

Feature extraction is an estimate of a variable from another set of variables (for example, observed speech signal time series), called a feature vector [1]. Feature selection is to convert these observation vectors into feature vectors. The goal of feature selection is to find transformations of relatively low-dimensional feature spaces, to retain information about the application, and to make meaningful comparisons using simple similarity measurements.

Speech parametrization is the conversion of speech signals into a set of feature vectors. The purpose of this transformation is to obtain a more compact, less redundant new representation, more suitable representation for statistical modelling and calculation of distance or any other type of score. The majority of the speech parameters used by the speaker verification system depend on the creep representation of the speech rate.

2.1 Filterbank-based cepstral parameters

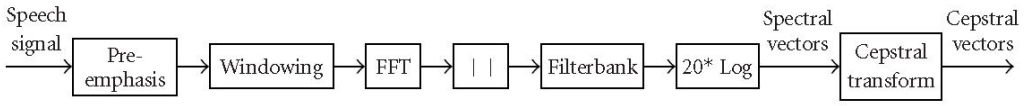

Figure 2.1: Modular representation of a filterbank-based cepstral parametrization.

Figure 2.1 shows a modular representation of the cepstrum representation based on the filterbank.

Typically, before a further step, the frame is pre-emphasized and multiplied by a smooth window function. The goal of the filter is to increase the high frequency of the spectrum, which is usually reduced by the speech production process. The pre-emphasis signal is obtained by applying the following filters:

$$x_p = x(t) - a \cdot x(t-1) \tag{2.1}$$

The value of a is usually obtained at interval 0.95 [0.98]. This filter is not always applicable, and some people prefer not to pre emphasize signal processing before. On the other hand, the window function (usually Hamming) is necessary due to the finite length of the DFT. The window is first applied to the beginning of the signal, and then further moved, and so on, until the signal arrives. The window provides a spectrum vector for each application of a part of the speech signal (after the application of FFT - see below). You

must set two quantities: the length of the window and the displacement between two consecutive windows. For window length, two values are the most commonly used: 20 milliseconds and 30 milliseconds. Hanning Hamming and Windows are the most commonly used speakers. Usually, Hamming window or Hanning window is used instead of rectangular window to reduce the original signal, thus reducing the side effect.

Once the speech signal is windowed and pre emphasized, the FFT is calculated. The famous fast Fu Liye transform (FFT) has been widely used in practice because of its simplicity and efficiency, and it can quickly segment the signal into frequency components. Usually only the amplitude spectrum is preserved because there is almost no emotional importance in the belief phase. After selecting the FFT algorithm, the only parameter used in the FFT calculation is the number of points. The number N is usually a power of 2, greater than the number of points in the window, usually 512. It extracts the modulus of FFT and obtains power spectrum and sampling 512 points.

DFT amplitude spectrum, that is, the overall shape of the spectral envelope, contains the information of the resonance characteristics of the channel, and is considered to be the largest part of the information in the ASV spectrum. The simple spectral envelope model uses a set of band-pass filters to integrate the energy of adjacent bands. The reason for spectral smoothing is that the size of the spectral vector decreases. In order to achieve this smoothing and obtain the envelope of the spectrum, we multiply the spectrum of the previously obtained filter banks. A filter bank is a series of bandpass frequency filters obtained by multiplying the spectrum to obtain a

specific average frequency band. The filter banks are defined by the shape and frequency of the filter (the left frequency, the center frequency and the right frequency). Filters can be either triangular or other shapes, and their frequency ratios may be different. In particular, some authors use the bark / Mel level to perform the frequency localization of the filter. This ratio is the auditory scale, analogous to the frequency scales of the human ear. The central frequency of the filter is given by the following equation

$$f_{MEL} = 1000 \cdot \frac{log(1 + f_{LIN}/1000)}{log2} \tag{2.2}$$

We use the logarithm of the spectral envelope and multiply each coefficient by 20 to obtain a spectral envelope in dB. At the processing stage, we obtain the spectral vector.

An additional transform called cosine discrete transform is usually applied to the spectral vector in speech processing and produces a cepstral coefficient

$$c_n = \sum_{k=1}^{K} S_k \cdot cos[n(k - \frac{1}{2})\frac{\pi}{K}], n = 1, 2, \cdots, L \tag{2.3}$$

where K is the number of log-spectral coefficients calculated previously, S_k are the log-spectral coefficients, and L is the number of cepstral coefficients that we want to calculate ($L \leq K$).

Once the cepstral coefficients are calculated, they can be centered, that is, subtracting the cepstral average vectors from each cepstral vector. This operation is called cepstral mean subtraction (CMS), which is usually used for speaker verification. The motivation of CMS is to remove slowly varying

convolution noise in cepstrum. The cepstral vector can also be reduced, that is, the variance is normalized to one component.

After the cepstral coefficients have been calculated, and possibly centred and reduced, we also incorporate in the vectors some dynamic information, that is, some information about the way these vectors vary in time. This is classically done by using the Δ and $\Delta\Delta$ parameters, which are polynomial approximations of the first and second derivatives

After calculating the cepstral coefficients, which may be concentrated and reduced, we also include some dynamic information in the vector, that is to say, these vectors change some information in time. This is the classical method, which uses the Δ and $\Delta\Delta$ parameters, which are polynomial approximations of the first and second derivatives.

$$\Delta C_m = \frac{\sum_{k=-l}^{l} k \cdot C_{m+k}}{\sum_{k=-l}^{l} |k|}, \Delta\Delta C_m = \frac{\sum_{k=-l}^{l} k^2 \cdot C_{m+k}}{\sum_{k=-l}^{l} |k^2|} \tag{2.4}$$

In this step, you can choose whether to log energy and delta delta energy into the feature vector. In practice, the former is often discarded, while the latter is retained.

Once you've calculated all the eigenvectors, the last very important step is to determine which vectors are useful. One way to look at the problem is to determine the vector corresponding to the speech part of the signal. Correspond to silence or background noise. A double Gauss model for computing eigenvector distribution. In this case, the lowest average Gauss corresponds to the background noise, while the highest average Gauss corresponds to the

speech part. Then Gauss's possibilities and silent background noises were abandoned by Gauss. A similar approach is to use the double Gauss model to calculate the logarithmic energy distribution for each speech segment, and apply the same principle.

Chapter 3

Speaker model and classifier

The main progress of speaker verification research is the improvement of classifier domain. Using vector quantization (VQ) and dynamic time warping (DTW) method, the original speaker verification system is developed. Subsequently, with the introduction of the Gauss mixture model (GMM) [4] [5], channel compensation and data variability have attracted more and more attention in the past twenty years of ASV research. An independent significantly improved generalized background model (UBM) based on GMM is proposed, and GMM is trained by maximum likelihood method. Another new paradigm of ASV technology is introduced by latent variable method. For example, a simulation method based on factor analysis (FA) for inter media variability of GMM hypermedia is proposed. Due to the combination of FA (JFA) success, i.e., the speaker factor as a direct classification feature, Dehak et al introduced the single integral subspace model of speaker, speaker and channel JFA in different subspaces. The recent speaker verification tech-

niques focus on the total variability modeling, also known as the I vector. My vector space is composed of a single speaker and a Gauss probability LDA correlation subspace channel model (further gplda), and the method effectively solves the coherent variation. The current ASV technique uses this I vector method, which provides an elegant framework for obtaining fixed length variable length speech statements. In recent years, deep learning has attracted wide attention and has aroused wide interest. Speaker verification, this study uses DNN model to train speech recognition, and build UBM, such as acoustic model, so that the rich information of mobile phones can be used to develop more effective background model. DNN has also successfully realized the extraction of speaker information features. [11] [10]

3.1 GMM-UBM system

The GMM-UBM system is a straightforward generative approach for ASV task, which was proposed in [5]. In this framework, training phase is preceded by estimation of a speaker-independent universal background model (UBM), using a sufficiently large speech data of several hours from multiple sources.Each speaker is represented as a GMM derived by maximum-a-posteriori (MAP) adaptation from UBM. For this purpose first, sufficient statistics of the features from speaker's enrolment utterances are computed. Then relevance MAP approach is used to estimate the weights, means and covariances of the target speaker model. During test or verification, average log-likelihood ratio is estimated using feature vectors from T speech frames

of test utterance against both target speaker model and the UBM.

3.1.1 Gaussian Mixture Models

The Gauss mixture model is usually used for acoustic learning tasks such as speech / speaker recognition, because it describes the different distributions of all feature vectors. GMM assumes that the characteristic vector x belonging to the model has the following probability:

$$p(x|w_i, \mu_i, \Sigma_i) = \sum_{i=1}^{K} w_i \mathcal{N}(x|\mu_i, \Sigma_i) \tag{3.1}$$

where

$$\mathcal{N}(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma_i|}} exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right) \tag{3.2}$$

subject to

$$\sum_{i=1}^{K} w_i = 1 \tag{3.3}$$

Therefore, GMM is only a weighted combination of multivariate Gaussian distributions, assuming that the eigenvectors are independent. (In fact, we use the diagonal covariance matrix eigenvector of the dimension, naturally independent of each other). GMM can use multiple clusters to describe the distribution of feature vectors, as shown in the figure.

GMM training process is μ_i, Σ_i, w_i, finds the best parameters and makes the model fit the maximum likelihood of all training data. More specifically, the expectation maximization (EM) algorithm is used to maximize the like-

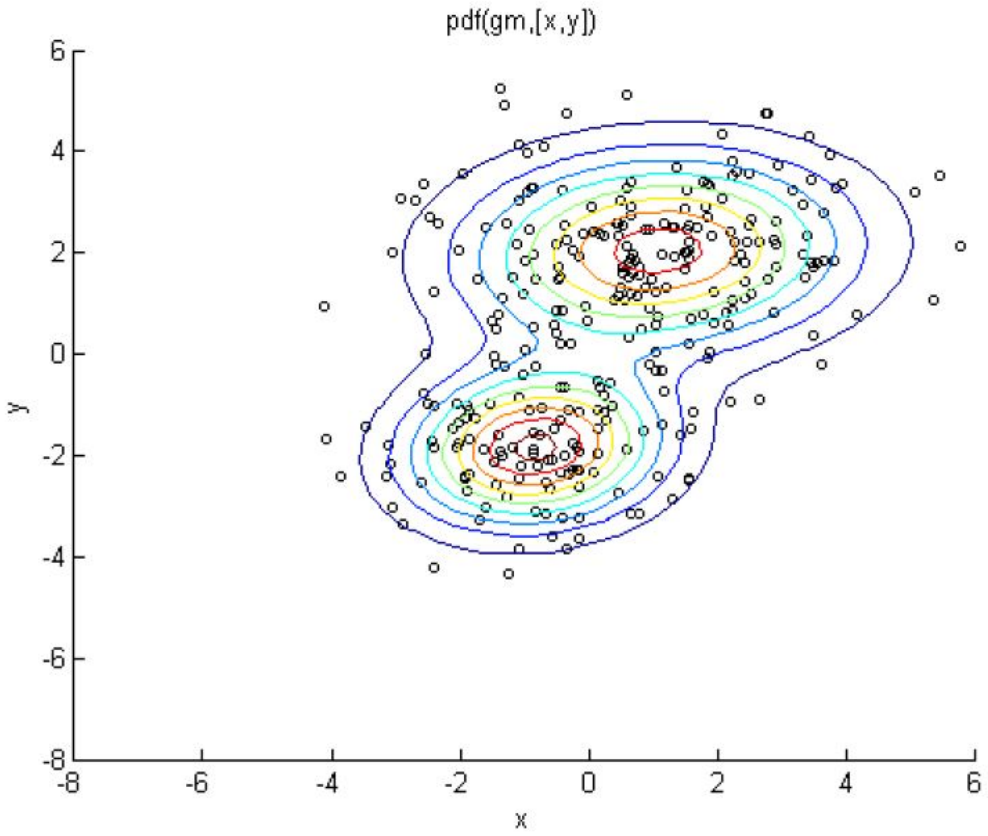

Figure 3.1: A Two-Dimensional GMM with Two Components.

lihood. In the case of GMM training, the two steps of an iteration of the algorithm are

- **E-Step** The probability of each data point (eigenvector) is estimated for each Gaussian to generate it. This is done directly by using the equation 3.1

- **M-Step** Modify the GMM parameters to maximize the likelihood of data. Here, the hidden variable z_{ij} is introduced to indicate where the i-th data point is generated by Gaussian j. It can be seen that instead of maximizing the possibility of data, we can maximize the likelihood of data relative to Z log.

  let $\theta = \{w, \theta, \Sigma\}$, the log likelihood function is

$$Q(\theta^{'}, \theta) = \mathbf{E}_Z[logp(X, Z)|\theta] \tag{3.4}$$

  where θ is current parameters, and $\theta^{'}$ is the parameters we are to estimate. Incorporating the constraint $\sum_{i=1}^{K} w_i = 1$ using Lagrange multiplier gives

$$J(\theta^{'}, \theta) = Q(\theta^{'}, \theta) - \lambda \left(\sum_{i=1}^{K} w_i = 1 \right) \tag{3.5}$$

  Set derivatives to zero, we can get the update equation

$$Pr(i|x_j) = \frac{w_i \mathcal{N}(x_j|\mu_j^{'}, \Sigma_j^{'})}{\sum_{k=1}^{K} w_k \mathcal{N}(x_k|\mu_k^{'}, \Sigma_k^{'})} \tag{3.6}$$

$$n_i = \sum_{j=1}^{N} Pr(i|x_j) \tag{3.7}$$

$$\mu_i = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_j) x_j \tag{3.8}$$

$$\Sigma_i = \left(\frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_j) diag(x_j x_j^T) \right) - diag(\mu_i^{'} \mu_i{}' T) \tag{3.9}$$

$$w_i = \frac{n_i}{N} \tag{3.10}$$

Although the general model supports the covariance matrix, i.e., it has all the elements of the covariance matrix, but only the diagonal covariance matrix is used in this paper. This is for three reasons. First, the large order diagonal covariance GMM can be used to model the density of m order covariance GMM. Second, the diagonal matrix GMM is more efficient than the full covariance GMM used in the training, because there is no need to repeatedly reverse the dollar D * D matrix. Again, from our observations, the diagonal matrix GMMs is superior to the full matrix GMM experience.

GMM can be considered as a mixture of parametric and non-parametric density models. Like the parametric model, it has the structure and parameters that control the density behaviour in a known manner, but without constraints, and the data must be of a specific distribution type, such as Gauss or Laplace operators. Like non parametric models, GMM has many degrees of freedom and can be modelled in any density without the need for excessive computation and storage. It can also be considered as the ergodic Gauss observation HMM with a single state HMM with Gauss mixed observation density, or with fixed equal transition probabilities. Here, the Gauss component can be viewed as a potentially wide range of speech feature models, a person's voice.

Using the GMM likelihood function is computationally simple, based on a well-known statistical model, sensitive to text independent tasks, not speaking time, and only the acoustic observations of the underlying distribution model. The latter has not yet utilized the higher level information of loudspeakers transmitted in time voice signals.

3.1.2 Universal Background Model

The general background model is a GMM model for training large numbers of speakers. Therefore, it describes the common acoustic features of human sound.

In the GMM-UBM system, we use a single, independent speaker representing the background model [5]. $p(X|\lambda_{\overline{hyp}})$. UBM is a trained and representative speaker independent speaker distribution model. Specifically, we need to select speech that can replace speech in the process of recognition. This applies not only to the type and quality of speeches, but also to the composition of speakers. For example, in the NIST-SRE single speaker detection test, it is known as a priori speech of local and long-distance calls, and male virtual speakers can only test male utterances. In this case, we used male telephone voice training to test UBM for men. In the absence of prior knowledge of the gender composition of alternative speakers, we will adopt gender independent language training.

3.1.3 Adaptation of Speaker Model

The vectors are represented independently by the background model. When the new speaker is registered in the system, the parameters of the background model adapt to the characteristic distribution of the new speaker. The adaptation model is then used as the speaker model. In this way, the model parameters will not be estimated from zero.Using existing knowledge (general speech data). The practice shows that it is beneficial to cultivate two independent background patterns: one is the female mode, the other is the male mode. Then, the new speaker model will be adjusted from the background model with the same gender as the new speaker.

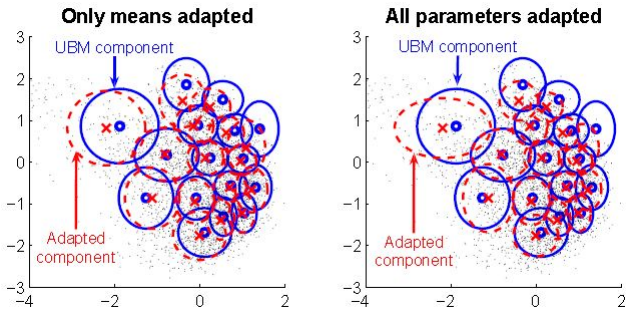

Figure 3.2: An example of GMM adaptation using the maximum a posteriori (MAP) principle. The Gaussian component of the generic background model (entity ellipse) applies to the training data (point) of the target speaker to produce a speaker model (virtual ellipse).

The basic idea of the adaptive method is to update the speaker's model by updating the trained parameters in the UBM. This provides a closer

coupling between the speaker model and the UBM, which not only produces better performance than the decoupling model, but also allows fast scoring techniques.

The specifics of the adaptation are as follows. Given a UBM and training vectors from the hypothesized speaker, $X = \{x_1, \cdots, x_T\}$, we first determine the probabilistic alignment of the training vectors into the UBM mixture components. That is, for mixture i in the UBM, we compute

The specific circumstances of adaptation are as follows. From the hypothesized speaker and UBM we train the training vector, $X = \{x_1, \cdots, x_T\}$, the training vectors are first determined by the probabilistic alignment of the UBM hybrid components. That is, for mixed I in UBM, we compute

$$Pr(i|x_t) = \frac{w_i p_i(x_t)}{\sum_{j=1}^{M} w_j p_j(x_t)} \tag{3.11}$$

We then use $Pr(i|x_t)$ and x_t to compute the sufficient statistics for the weight, mean, and variance parameters:

$$n_i = \sum_{t=1}^{T} Pr(i|x_t) \tag{3.12}$$

$$E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_t) x_t \tag{3.13}$$

$$E_i(x^2) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_t) x_t^2 \tag{3.14}$$

Finally, use the new enough statistics for the training data to update the sufficient statistics of the old UBM of Mixture i to create the adaptation

parameter for Mixture i with the following equation:

$$\widehat{w}_i = [\alpha_i^w n_i/T + (1-\alpha_i^w)w_i]\gamma \tag{3.15}$$

$$\widehat{\mu}_i = \alpha_i^m E_i(x) + (1-\alpha_i^m)\mu_i \tag{3.16}$$

$$\widehat{\sigma}_i^2 = \alpha_i^v E_i(x^2) + (1-\alpha_i^v)(\sigma_i^2 + \mu_i^2) - \widehat{\mu}_i^2 \tag{3.17}$$

The adaptation coefficients controlling the balance between old and new estimates are $\alpha_i^w, \alpha_i^m, \alpha_i^v$for the weights, means and variances, respectively. The scale factor, γ, is computed over all adapted mixture weights to ensure they sum to unity. Note that the sufficient statistics, not the derived parameters, such as the variance, are being adapted.

The adaptation coefficients that control the balance between old and old estimates are $\alpha_i^w, \alpha_i^m, \alpha_i^v$,weight, mean and variance, respectively. Calculate the scale factor , γ,on the weight of all the mixture to ensure that they are uniform. Note that you are adjusting enough statistics, rather than exporting parameters such as variance.

In the recognition mode, the MAP adaptation model and the UBM are coupled, and the recognizer is usually considered to be *Gaussian mixture model - universal background model*, or simply GMM-UBM. The match score depends on both the target model λ_{target} and the background model λ_{UBM} via the average log likelihood ratio:

$$LLR_{avg}(X, \lambda_{target}, \lambda_{UBM}) = \frac{1}{T}\sum_{t=1}^{T} logp(X_t|\lambda_{target}) - logp(X_t|\lambda_{UBM}) \tag{3.18}$$

It basically measures the difference between the target and the background model when generating observations.

3.2 Support vector machine using GMM supervector

Support vector machine (SVM) is a powerful discriminative classifier which is widely used in speaker recognition in recent years. Support vector machine is a natural way to solve this problem, basically because of the recognition of the speaker is two kinds of problems, so we must be in the crowd between the speaker or speaker to make assumptions to make assumptions. Support vector machines perform nonlinear mapping from input space to SVM feature space. The linear classification technique is then applied to this potential high-dimensional space. The main design part of the support vector machine is the kernel, which is the inner product of the support vector machine feature space. The basic goal of support vector machine design is to find the appropriate metric in the SVM feature space associated with the classification problem because of the distance metric produced by the inner product and vice versa.

An SVM is a two-class classifier constructed from sums of a kernel function $K(\cdot, \cdot)$

$$f(x) = \sum_{i=1}^{L} \alpha_i t_i K(x, x_i) + d \tag{3.19}$$

where the t_i are ideal outputs, d is a learned constant, $\sum_{i=1}^{L} \alpha_i t_i = 0$, and

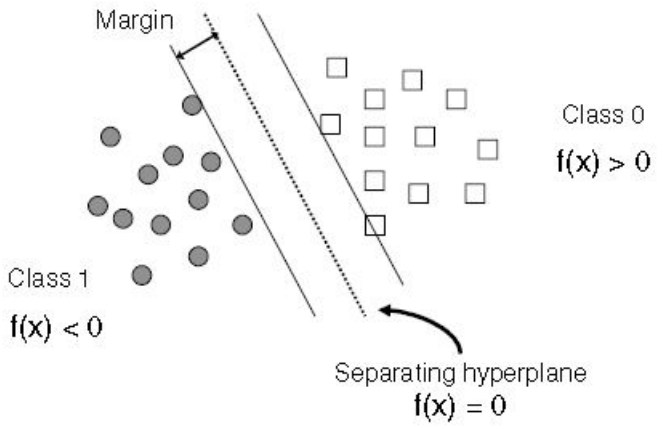

Figure 3.3: Support vector machine concept.

$\alpha_i > 0$. Creative x_i is a support vector and is obtained from the training set through the optimization process. According to any support support, the ideal game is 1 or -1, respectively 0 or 1 level. The basis of a class decision is that $f(x)$ is above or below the threshold.

The kernel $K(\cdot, \cdot)$ is constrained to have certain properties (the Mercer condition), so that $K(\cdot, \cdot)$ can be expressed as

$$K(x, y) = b(x)^t b(y) \tag{3.20}$$

where $b(x)$ is a mapping from the input space (where x lives) to a possibly infinite dimensional expansion space. The Mercer condition ensures that the margin concept is appropriate, and the optimization of the SVM is well defined.

The optimization conditions depend on the concept of maximum margin.

For separable datasets, the system places the hyperplane in the high dimensional space so that the hyperplane has the maximum margin. The data points from the training set located on the boundary are support vectors. Then the focus of the SVM training process is to establish the boundaries between classes. [6]

3.3 I-vector system

In recent years, the development of speaker recognition technology has successfully implemented a system based on the low dimensional representation of speech segments, called identity vectors or I vectors [7]. The vector is a compact representation of the Gauss mixture model (GMM) hyper vector, which captures most of the changes in the super Gauss hyper vector. It is an average graph obtained by a posteriori distribution estimation method.

3.3.1 Total Variability

The classical joint factor analysis modelling based on speaker and channel factors consists of defining two different spaces: The eigen space matrix of the speaker space V and the defined channel space defined by the U system eigen channel matrix. The method we propose is based on defining a space, not two separate spaces. This new space, which we call the total variation space, includes the variations of the speaker and the channel. It is defined by the total variance of the eigenvalues of the largest eigenvector matrix which contains the total covariance matrix of the corresponding variance. In the

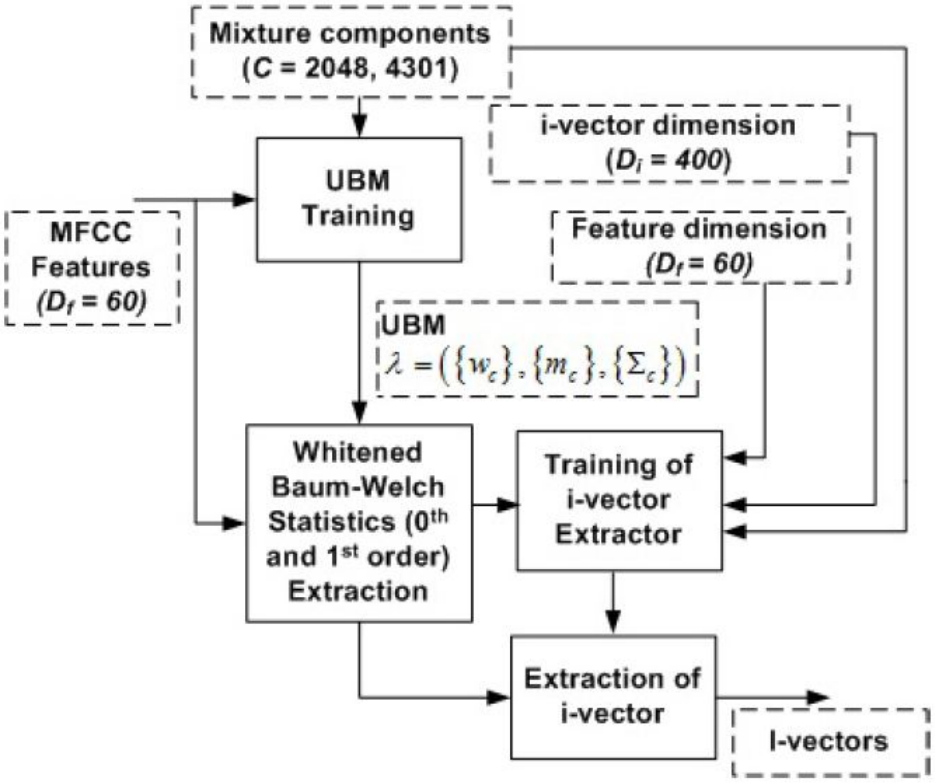

Figure 3.4: Block diagram showing different stages of I-vector extraction process.

new model, we do not distinguish the speaker effect and the channel effect in the super Gauss vector space. Given a language, new speakers and channels depend on the system GMM super vector M modified as follows:

$$M = m + Tw \tag{3.21}$$

Where m is the speaker vector and channel independent super vector (which can be considered UBM super vector), T is the lower rank rectangle matrix, w is the random vector of$\mathcal{N}(0, I)$ with standard normal distribution. The The component of the vector w is the total factor. We refer to these new carriers as identity carriers or simply i carriers. In this modelling, assuming that M is a normal distribution, the mean vector and the covariance matrix TT^t. The process of training the total variation matrix T is exactly the same as learning the intrinsic speech matrix V, except for an important difference: in the intrinsic speech training, all the recordings of a given speaker are considered to belong to the same person; however, In the case of a matrix, the entire set of utterances of a given speaker is thought to be produced by different speakers (we pretend that each utterance from a given speaker is generated by a different speaker). The new model we propose can be seen as a simple factor analysis that allows us to project speech discs to low-dimensional total variation space.

The total factor w is a hidden variable that can be defined by its posterior distribution for the Baum-Welch statistic for a given speech. The posterior distribution is a Gaussian distribution with an average of the dis-

tribution that corresponds exactly to our i-vector. Similar to the Baum-Welch statistics extracted using UBM. Suppose we have a series of L frames $\{y_1, y_2, \cdots, y_L\}$ and a UBM Ω consisting of C mixed components defined by some feature space of dimension F. In order to study the basic total variability subspace, we need to calculate the Baum-Welch statistic defined as

$$N_k(s) = \sum_t \gamma_{tk}(s) \tag{3.22}$$

$$F_k(s) = \sum_t \gamma_{tk}(s) O_t(s) \tag{3.23}$$

Where $N_k(s)$ and $F_k(s)$ represent the zero and first order statistics of the speech session s respectively, where $gamma_{tk}(s)$ is the posterior probability of the mixed component k Given the observation vector $O_t(s)$ at time frame t.

The i vector for a given speech can be obtained using the following equation:

$$w = (I + T^t \Sigma^{-1} N(u) T)^{-1} T^t \Sigma^{-1} F(u) \tag{3.24}$$

We define $N(u)$ as a diagonal matrix of dimension $CF\mathbf{x}CF$ whose diagonal blocks are $N_k I$. is a supervector of dimension $CF\mathbf{x}1$ obtained by concatenating all first-order BaumWelch statistics F_c for a given utterance. Σ is a diagonal covariance matrix of dimension $CF\mathbf{x}CF$ is estimated during factor analysis training, and the residual variation of the total variation matrix T is not simulated.

3.4 PLDA classifier

Linear dimensionality reduction methods, such as LDA, are often used in object recognition for feature extraction, but do not address the problem of how to use these features for recognition. The latent variables of PLDA represent both the class of the object and the view of the object within a class. The usual LDA features are derived as a result of training PLDA, but in addition have a probability model attached to them, which automatically gives more weight to the more discriminative features. With PLDA, we can build a model of a previously unseen class from a single example, and can combine multiple examples for a better representation of the class.

Probabilistic LDA is a general method that can accomplish a wide variety of recognition tasks, which was first proposed in face recognition area [8]. In one-shot learning, a single example of a previously unseen class can be used to build the model of the class. Multiple examples can be combined to obtain a better representation of the class. In hypothesis testing, we can compare two examples, or two groups of examples, to determine whether they belong to the same (previously unseen) class. This can further be used to cluster examples of classes not observed before, and automatically determine the number of clusters.

Linear discriminant analysis (LDA) is a technique that models both intra-class and inter-class variance as multidimensional Gaussians. It seeks directions in space that have maximum discriminability and are hence most suitable for supporting class recognition tasks. In this section we present a

probabilistic approach to the same problem which we term probabilistic LDA or PLDA. The relationship between PLDA and standard LDA is analogous to that between factor analysis and principal components analysis.

We assume that the training data consists of J images each of I individuals. We denote the jth image of the ith individual by x_{ij} . We model data generation by the process:

$$x_{ij} = \mu + Fh_i + Gw_{ij} + \xi_{ij} \tag{3.25}$$

This model comprises two parts: (i) the signal component $\mu + Fh_i$ which depends only on the identity of the person but not the particular image (there is no dependence on j). This describes between-individual variation. (ii) the noise component $Gw_{ij} + \xi_{ij}$ which is different for every image of the individual and represents within-individual noise.

The term μ represents the overall mean of the training dataset. The columns of the matrix F contain a basis for the between-individual subspace and the term h_i represents the position in that subspace. The matrix G contains a basis for the within-individual subspace and w_{ij} represents the position in this subspace. Remaining unexplained data variation is explained by the residual noise term ξ_{ij} which is defined to be Gaussian with diagonal covariance Σ.

In the parlance of factor analysis, the matrices F and G contain factors and the latent variables h_i and w_{ij} are factor loadings. For readers familiar with LDA, the columns of F are roughly equivalent to the eigenvectors of

the between-individual covariance matrix, and the columns of G are roughly equivalent to the eigenvectors of the within individual covariance matrix. The term h_i is particularly important as this represents the identity of individual i. We term this a *latent identity variable*: in recognition we will consider the likelihood that two face images were generated from the same underlying h_i.

More formally, we can describe the model in Equation 3.25 in terms of conditional probabilities:

$$Pr(x_{ij}|h_i, w_{ij}, \theta) = \mathcal{G}_x[\mu + Fh_i + Gw_{ij}, \Sigma] \tag{3.26}$$

$$Pr(h_i) = \mathcal{G}_h[0, I] \tag{3.27}$$

$$Pr(w_i j) = \mathcal{G}_w[0, I] \tag{3.28}$$

where $\mathcal{G}_a[b, C]$ represents a Gaussian in a with mean b and covariance C.

3.4.1 Gaussian PLDA (G-PLDA)

Assuming R speech for the speaker, the corresponding i-vector collection is expressed as $\{\eta_r : r = 1, \cdots, R\}$.Then, the introduced G-PLDA model [9] assumes that each i-vector can be decomposed into

$$\eta_r = m + \phi\beta + \Gamma\alpha_r + \epsilon_r \tag{3.29}$$

The terms identified by the speaker, the model consists of two parts: the specific part of the descriptor, which describes the variability between the speakers and does not depend on the particular discourse; channel components. This is discourse dependent and describes the differences between

the speakers. In particular, m is a global offset; provides the basis for the speaker's specific subspace (functional lecture); A potential identity vector with a standard normal distribution. The column of ; A potential vector with a standard normal distribution; and is the Gaussian residual of the mean and diagonal covariance assumed to be zero. In addition, it is assumed that all potential variables are statistically independent. Since the i vector I handle in this work has a sufficiently small dimension (ie, our experiment is 400), and assuming is a complete covariance matrix, and removes the feature channel. Thus, the improved G-PLDA model used herein is as follows:

$$\eta_r = m + \phi\beta + \epsilon_r \tag{3.30}$$

Using the EM algorithm to obtain the ML point estimation of model parameters from a large number of development data sets.

3.4.2 Verification score

For the speaker verification task, given two i vectors η_1 and η_2 for trial, we are interested in testing two alternative assumptions: $\mathcal{H}_\int$, η_1 and η_2 Share the same speaker with the potential variable β or $\mathcal{H}_\lceil$ and generate the i vector using the different identity variables β_1 and β_2. The verification score can now be calculated as the log likelihood ratio of the hypothesis test

$$score = log\frac{p(\eta_1, \eta_2|\mathcal{H}_s)}{p(\eta_1|\mathcal{H}_d)p(\eta_2|\mathcal{H}_d)} \tag{3.31}$$

3.5 Deep neural networks for extracting baum welch statistics

In the field of speech recognition, deep level neural networks (DNN) have recently been successfully applied to acoustic modelling to achieve large improvements compared to standard GMM models [11]. Neural network is a standard forward neural network, which is larger than the hidden layer of traditional neural network (there are thousands of nodes in each hidden layer) and depth (5-7). Neural network training commonly used standard discrimination BP algorithm and stochastic gradient descent method.

The posterior probability, as the speaker's shape and space, is extracted from frame alignment factor and UBM is used to train the speaker in the standard frame of instruction modeling, DNN. The use of speech perception model is due to the influence of speech content on speech signals, which has been neglected in text independent speaker verification.

DNN replaces GMM framework with different types of models after calculation. In the GMM model, it is from the Gauss mixture model of a person, in the circumstances of the neural network is the senones (Asia) decision tree automatic speech recognition using a standard. The posterior probabilities are computed at the standard, and they enter the state-of-the-art models of vector / PLDA that are zeroth order and first order statistics [10]. An attractive advantage of this approach is that the characteristics of frame alignment and sufficient statistical data can be changed because the two processes are

now effectively decoupled. Therefore, the system can use the optimization function to maximize the recognition and calculation of frame alignment using mobile phone speaker recognition accuracy, to obtain the I vector and the last speaker to confirm the results of the best features of sufficient statistics.

3.5.1 DNNs for ASR

In the most advanced ASR system, the pronunciation of all words is represented by a series of farmer Q (e.g., bound states). Each morpheme is used to simulate the state of a group of three-phase near acoustic spaces. Usually, the automatically defined morpheme group Q uses the maximum likelihood (ML) decision tree method. A decision tree is designed to increase maximum likelihood growth by requiring a set of local optimal problems, assuming that the data on each side of the split can be modeled with the Gauss model. The decision tree leaves and then as senones eventually set.

The Viterbi decoder is used to adjust the training data to the corresponding senones. These routes are used to estimate the probability distribution of P (x — on Q), where X is the training data and the Q observation vector is morpheme. The estimation and adjustment of the observed probability distribution are optimized by iteration and iteration. Traditionally, a GMM is allocated. In the recent system, one is used to estimate the posterior probability of the acoustic characteristics of DNN. Probabilities can be observed from the Sen agricultural rules using Bayesian ness, obtained before, as follows:

$$p(x|q) = p(q|x)p(x)/p(q) \tag{3.32}$$

Where $p(x|q)$ is the observed probability of decoding, $p(q)$ is the previous senone, and $p(q|x)$ is the senone posterior from DNN. Figure 1 shows a flow chart for training the DNN for ASR. A pre-trained Hidden Markov Model (HMM) ASR system with GMM status is required to produce a subsequent DNN training match. The final acoustic model consists of the original HMM of the previous HMM-GMM system and the new DNN.

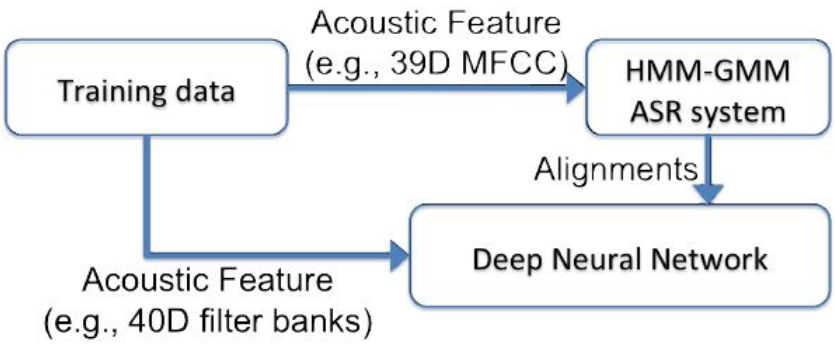

Figure 3.5: The flow diagram for training a DNN for ASR.

3.5.2 A DNN/i-vector framework

The need for a speech signal model to maximize the signal model of each class means a spatial change factor. The classes defined by UBM do not have intrinsic meaning. Each Gaussian covers only the feature space, which may include a portion (or three) of a different mobile phone instance, rather

than a single or even a specific phoneme. If a person says some phonemes, say / acrylic / is very different, and the corresponding speech frame may be associated with other phonemes training Gaussian alliance, say / Australia / ... so it is necessary to adapt to the corresponding / acrylic / gaussian displacement speaker sound / Acrylic / Shock. Only the communication / AO / Gaussian means will affect these frames. The final spatial factor will not contain the fact that the speaker is pronounced / very different from other information.

On the other hand, when calculating the exact posture probability of the corresponding spelling pepper and predicting the correct frame of the Cypriot, it is used to estimate the change in the way each morpheme. In the above example, the corresponding change / acrylic / frame will be assigned to the correct morpheme and means for those Cypriot results. The i-vector will reflect the fact that the speaker is very different from the general population. Simply put, when we can put this "apple" speech Saverman definition: each frame is the same phoneme training frame content comparison.

In acoustic modeling, DNNs have been shown to be superior to GMM models because they use longer context windows and differentiated cultures. As a result, a DNN model gives a better morpheme estimate than the supervised UBM. Note that an important feature of our approach is that there is no need to design a compromise that is effective for both ASR and SID. In fact, the neural network system can use completely different features for speaker identification, as long as it can improve the posterior probability estimation. This is similar to using a single center alignment in a multi-feature

SID system, rather than aligning it with a feature. The proposed DNN / space factor hybrid framework flow chart.

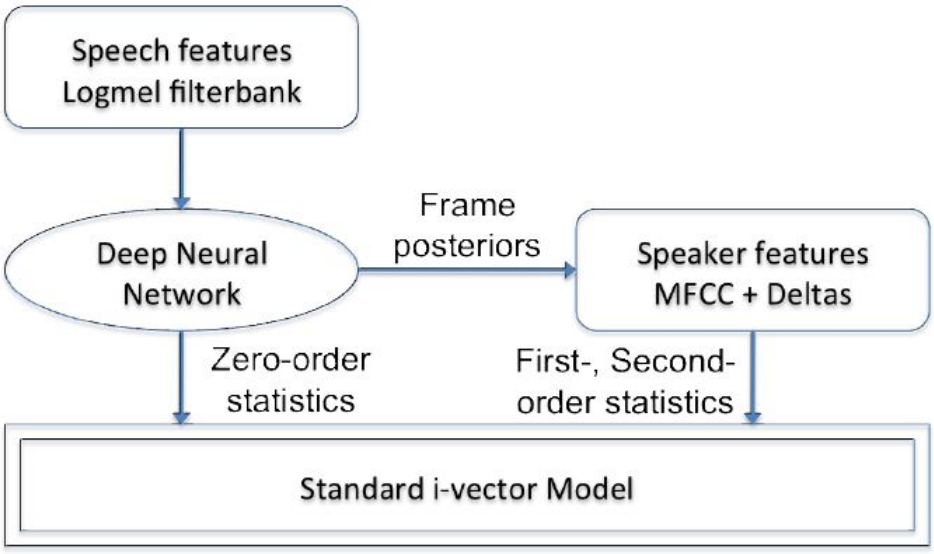

Figure 3.6: The flow diagram of the DNN/i-vector hybird farmework.

Chapter 4

Normalization technique

4.1 Feature normalization

In general, general noise suppression techniques can be used to improve the quality of the original time domain signal before feature extraction. However, as an additional step in the overall identification process, the signal enhancement will increase the computational load. The design requires a robust feature extractor, or a feature extraction before modeling or normalizing the matching algorithm.

The simplest way to normalize a feature is to subtract the average of each feature in the entire speech. In the logarithmic spectrum and the cepstrum region, the convolution channel noise becomes an addition. By subtracting the mean vector, the feature set obtained from two different channels is transformed into zero mean value, which reduces the influence of channel. Similarly, the variance of the feature can be balanced by dividing each feature

by its standard deviation. When VAD is used, normalized statistics are usually calculated based on the detected speech frame.

Discourse Mean and Variance Standardization assumes that the channel effect is constant throughout the discourse. In order to relax the assumptions, the mean and variance estimates can be updated by sliding the window. The window should be long enough to well estimate the mean and variance, but it is short enough to capture the time-varying properties of the channel. The typical window size is 3-5 seconds.

The characteristic deformation and short term Gaussian purpose is to modify the short term feature distribution to refer to the distribution. This is achieved by the cumulative distribution function of the "twist" feature, which makes it match the reference distribution function, such as Gaussian. Each feature stream is independently deformed. By applying a global linear transformation before warping, the independence hypothesis is relaxed, with the aim of achieving short-term decorrelation or feature independence. Although it is observed that the Gaussian characteristic improves the accuracy of the characteristic warpage, it is quite complicated to implement.

RASTA filtering uses a bandpass filter in a logarithmic or cepstrum domain. The filter is applied along the time trajectory of each feature and suppresses the modulation frequency outside the typical speech signal. For example, a slowly changing convolution channel noise may be considered as a low frequency portion of the modulation spectrum. Note that the RASTA filter is signal independent, whereas the CMS and variance normalization are adaptive because they use the statistics of the given signal.

4.2 Score normalization

The last step in speaker verification is decision making. This procedure involves comparing the likelihood between the desired speaker model and the incoming speech signal and the decision threshold. If the likelihood is higher than the threshold, the claimed speaker will be accepted or rejected [12].

Decision threshold adjustment is very troublesome in speaker verification. If the selection of the numerical value is still an open problem in the domain (usually a fixed experience), the reliability of the system can not be guaranteed at runtime. This uncertainty is mainly due to the difference between the experimental results and the actual field.

This fraction varies from source to source. First of all, the nature of the registered material can vary from speaker to speaker. It can also be analyzed from the aspects of speech content, duration, environmental noise and speaker model training quality. Second, registration data (for speaker modeling) and test data may exist mismatches, which is the main problem in speaker recognition. Two main factors may lead to mismatch: speaker / speaker (emotional changes, internal change itself through the speaker changes caused by health status and age) of some environmental conditions and transmission channel, recording material or acoustic environment. On the other hand, the variability of human language is a special problem in speaker independent threshold system, and it is also a potential factor affecting the reliability of decision boundary. In fact, this change is the speaker can not be measured directly, it is not a simple speaker protection verifica-

tion system (by decision process) of all potential imposter attacks. Finally, as the training materials, the test value is divided into client and impostor impact test section quality characteristics.

Fractional normalization has been explicitly introduced to cope with changes in scores and make it easier to adjust speaker independent decision thresholds.

A score normalization of the form

$$s' = \frac{s - \mu_I}{\sigma_I} \tag{4.1}$$

Is the normalized s' is the normalized score, s is the original score, μ_I and σ_I are the estimated mean and standard deviation of the fake score distribution, respectively. In zero normalization ("Z norm"), impersonation vertex statistics μ_I and σ_I are related to the target speakers, and they are calculated off-line in the speaker registration phase. This is achieved by matching a batch of non-target statements with the target model and obtaining the mean and standard deviation of these scores. In the test normalization ("T-norm"), the parameters are test-dependent, which are calculated in the verification phase "in flight" by matching the eigenvector of the unknown loudspeaker with a set of colon models statistics.

4.2.1 Znorm

Zero-normalization (Znorm) technology has been used extensively for speaker verification in the mid-1990s. In practice, the speaker model is tested against

a set of speech signals generated by some impostors, resulting in impersonator similarity score distributions. The speaker-related mean and variance normalization parameters are estimated from the distribution (see (16)) Runtime Verification System produces similarity scores. One advantage of the Znorm is that the estimation of the normalized parameters can be performed during the speaker model training period.

4.2.2 Tnorm

Specification or parameter estimation of the mean and variance of the distribution of the impostor scores based on test specification, using the test speech signal is different from the Znorm rather than the impostor model. During the test, the input speech signal and that compared with the traditional speaker model and a set of fraud model to estimate the fraction distribution and normalized parameters of continuous impostor. If Znorm is a speaker normalization technique, Tnorm is a test dependency. Similarly, in the process of testing speech testing and standardized parameter estimation, Tnorm avoids the possible problems of znorm based on possible mismatches between test and standardized utterances. On the contrary, Tnorm needs to be tested online.

Chapter 5

Pre-process and dimension reduction

Linear dimensionality reduction method has a long tradition in object recognition. Most notably, these methods include principal component analysis (PCA) and linear discriminant analysis (LDA). Although PCA identifies the energy of most of the data in the linear subspace, LDA identifies the data of the different classes relative to the most extended subspace in each class. This makes LDA suitable for recognition, classification and other problems. One question that can not be answered by dimensionality reduction is how do we deal with the low dimensional representation of data? A common method is to project the data into the PCA subspace, thus eliminating the singularity, and then find the LDA subspace. However, after projection, how do we combine the multivariate representations of the resulting components? Obviously, some dimensions (for example, the main projection direction de-

termined by LDA) must be more important than others, but how do we incorporate this difference into the importance of identification? How do we perform tasks such as classification and hypothesis testing, and before we use multiple instances of a new class, we haven't seen these tasks yet.

5.1 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a generalization of Fisher linear discriminant. It is used in statistics, pattern recognition and machine learning to find, characterize, or separate linear combinations of two or more classes of objects or events. The obtained combinations can be used as linear classifiers, or more widely used to reduce the dimensionality before subsequent classification.

LDA is closely related to the analysis of variance (ANOVA) and regression analysis, and attempts to represent dependent variables as other features or linear combinations of measurements. However, the variance analysis of categorical variables and continuous variables, and discriminant analysis has continuous and categorical variables (i.e. category labels), Logistic regression and probit regression analysis of variance is more similar than LDA, because they also explained a categorical variable by continuous parameter. In the application of the irrational assumption of normal distribution of independent variables, these other methods are preferred, which is the basic assumption of the LDA method.

LDA is also closely related to principal component analysis (PCA) and

factor analysis, since they both seek linear combinations of the best data explanatory variables. LDA explicitly attempts to simulate the differences between classes of data. On the other hand, PCA doesn't take into account any differences in class. Factor analysis establishes a feature combination based on difference rather than similarity. Discriminant analysis is also different from factor analysis, because it is not an interdependent technique: independent variables and dependent variables (also known as standard variables) must be distinguished.

Linear Discriminant Analysis is commonly used to identify the linear features that maximize the between-class separation of data, while minimizing the within-class scatter. Consider a training data set containing N examples $\{x_1, \cdots, x_N\}$, where each example x_i is a column vector of length d. Each training example belongs to one of the K classes. Let C_k be the set of all examples of class k, and let $n_k = |C_k|$ be the number of examples in class $k = 1, \cdots, k$. In LDA, the within-class and between-class scatter matrices are computed:

$$S_w = \frac{\Sigma_k \Sigma_{i \epsilon C_k} (x_i - m_k)(x_i - m_k)^T}{N} \tag{5.1}$$

$$S_b = \frac{\sum_k n_k (m_k - m)(m_k - m)^T}{N} \tag{5.2}$$

where

$$m_k = \frac{1}{n_k} \Sigma_{i \epsilon C_k} \tag{5.3}$$

is the mean of kth class, and

$$m = \frac{1}{N} \sum_{i} x^i \tag{5.4}$$

is the mean of the data set. We seek the linear transformation

$$x \to W^T x \tag{5.5}$$

The difference between classes is maximized relative to the intraclass variance, where W is a $d\ textbf{}xd'$ matrix that is the required dimension. It can be seen that the optimal W column is a generalized eigenvector that makes

$$S_b w = \lambda S_w w \tag{5.6}$$

Corresponds to d maximum eigenvalues. One consequence of this result is that W is diagonalizing both the scattering scattering matrices $W^T S_b W$ and $W^T S_w W$. In other words, the LDA will be released between the class and the internal data. LDA projection can be obtained by fitting a Gaussian mixture model to the training data. The resulting hybrid model can be used to classify the categories represented in the training data rather than categorize them. For this purpose, different probabilistic models are needed, provided by Probabilistic LDA.

As mentioned earlier, the i-vector simulates the speaker and channel-related information in the same total variability subspace. Thus, in order to select the most relevant feature subset for the speaker recognition task, the LDA may be applied to the i vector to eliminate the direction in which the speaker identification is not information. In addition, reducing the dimen-

sion of the i-vector by LDA can improve the computational efficiency of the subsequent back-end components in the system.

There are three disadvantages to the properties of the dispersion matrix $S_b and S_w$. First, let's assume that the basic distribution of this class is Gaussian, that is, the covariance matrix for all categories. Therefore, it can not be expected that the parameter LDA will well fit non-Gaussian and multimode (rather than unimodal) distributions. As known in the speaker recognition community, the actual distribution of i vectors may not necessarily be Gaussian distributions. This is especially true when recording sound in the presence of noise and channel distortion. In addition, for NIST SRE-type scenarios, records come from various sources and are collected (sometimes outside), but there is no guarantee that the distribution is single. Second, note that the Sb level is C-1, which means that the parameter LDA can provide the most C-1 discrimination. However, in applications such as speech recognition where the number of language categories is much smaller than the size of the i-vector, this may not be enough. However, this may not pose a challenge to the speaker recognition task, where the number of trained speakers exceeds the dimension of the total variance subspace. Finally, since only the type centroid of Sb is considered, the parameter LDA can not effectively capture the boundary structure between adjacent categories, which is crucial for classification.

5.2 Non-parametric discriminant analysis

As we have said repeatedly, L-type classification only needs (L-1) features. However, (L-1) features are suboptimal in the Bayesian sense, unless the posterior probability function is chosen, although they are optimal for the criteria used. Therefore, if the estimation of Bayesian error in feature space is much larger than that in the original variable space, some methods must be designed to enhance the feature extraction process.

One possibility is to artificially increase the number of classes. So that we can increase the level of Sb. This can be done by dividing each class into multiple clusters. For the case of multi-modal behavior, clustering algorithms that "correctly" identify clusters can be found, which may work well. As a second possibility, after determining the L-1 features, they can be removed, leaving subspaces orthogonal to the extracted features. A similar process can then be applied to subspaces to extract additional features.

In this section, a non-parametric discriminant analysis is introduced to overcome the two problems mentioned earlier, the algorithm was proposed early in [13] [14] and further implemented in IBM speaker recognition system [15]. The basis for this expansion is the use of k-nearest (kNN) techniques to measure non-parametric interspersed interspersed scatter matrices between local base classes and are generally full-rank. As a result, neither artificial generation nor sequential methods are required. In addition, the non-parametric nature of the scatter matrix inherently causes the extracted features to retain structure important for classification.

The determination of the linear mapping can be thought of as finding the rotation and multiplication of the original data space, and then selecting the subspace in which all subsequent work is performed. Therefore, two issues must be addressed. First, you must specify the process of determining rotation and scaling. Second, you must specify the sort order of the rotation and zoom features to facilitate the selection process.

Non-parametric discriminant analysis (NDA) is introduced to overcome the parameter form of LDA by extending the scattering matrix. The normality assumption of LDA is relaxed, so it can deal with the abnormal data distribution, by combining the intra class scatter matrix and the inter class scatter matrix between the s_b direction and the boundary of s_w, respectively. In NDA, the expected value represents the global information of each class and the average value of the local sample instead of the nearest neighbour based on the individual sample. More specifically, in the NDA method, the inter class scatter matrix is defined as,

$$\overline{S_b} = \sum_{i=1}^{C} \sum_{i=1, j \neq i}^{C} \sum_{l=1}^{N_i} w_l^{ij} (x_l^i - \mathcal{M}_l^{ij})(x_l^i - \mathcal{M}_l^{ij})^T \tag{5.7}$$

where x_l^i denotes the l^{th} sample from class i, and $\mathcal{M}_l^{ij}$ is the local mean of k-NN samples for x_l^i from class j which is computed as,

$$\mathcal{M}_l^{ij} = \frac{1}{K} \sum_{k=1}^{K} NN_k(x_l^i, j) \tag{5.8}$$

where $NN_k(x_l^i, j)$ is the k^{th} nearest neighbor of x_l^i in class j.The weighting function $w_l^{ij}$is defined as

$$w_l^{ij} = \frac{min\{d^\alpha(x_l^i, NN_k(x_l^i, i)), d^\alpha(x_l^i, NN_k(x_l^i, j))\}}{d^\alpha(x_l^i, NN_k(x_l^i, i)) + d^\alpha(x_l^i, NN_k(x_l^i, j))} \tag{5.9}$$

where α is a constant between zero and infinity, and $d(\cdot)$ denotes the distance (e.g., cosine or Euclidean). The weighting function is a local gradient that emphasizes amplitude to reduce their influence on the scattering matrix. For samples near the classification boundary, the weight parameter is close to 0.5, and drops off to zero as we move away from the classification boundary. The control parameter, *alpha*, adjusts how rapidly w falls to zero as we move away.

The nonparametric within-class scatter matrix, $\overline{S_w}$, is computed in a similar fashion as $\overline{S_b}$, except the weighting function is set to 1 and the local gradients are computed within each class. The NDA transform is then formed by calculating the eigenvectors of $\overline{S_w}^{-1}\overline{S_b}$.

Careful study of non-parametric interspersed distribution matrix can make three important observations. First, we note that the local mean vector $\mathcal{M}_l^{ij}$ approaches the global mean class j (ie μ_j) as the nearest neighbor number K approaches N_j, . In this case, if we set the weight parameter to 1, the NDA transform essentially becomes an LDA projection, which means that LDA is a special case of the more general NDA.

Second, $\overline{S_b}$ is usually full-rank because all samples are considered in the scatter matrix between nonparametric classes (not just centroids). This means that, unlike LDAs that provide the most C-1 discriminant features, NDA usually results in a d-dimensional vector for classification (assuming

d-dimensional input space). As we discussed earlier, this is important for applications such as speech recognition, where the number of classes is much less than the total subspace (or general input space) dimension.

Finally, NDA is more effective than LDA in maintaining different categories and crossing different categories of complex structures (ie, local and boundary structures). As can be seen from the example (k is set to 1 for simplicity), LDA uses only the global gradient obtained from the two centroid levels to measure the scatter between classes. On the other hand, NDA uses a local gradient along the boundary highlighted by the weighting function w_l^{ij}. Therefore, the boundary information is embedded in the result transformation.

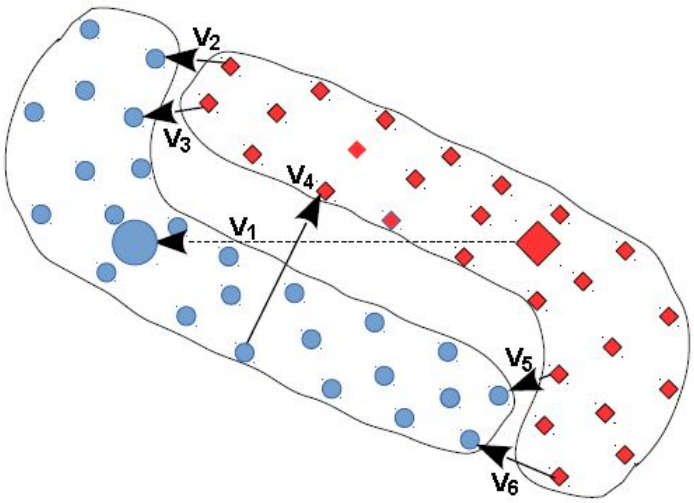

Figure 5.1: An example of a symbol with a non-parametric argument between two classes. v1 represents the global gradient of the class centroids.

From another class to the NN, each vector indicates the direction of the local to another class. If we select these vectors only from the samples located at the classification boundaries (V1, V3, V4, V5, etc.) then the scatter matrix of these vectors should specify the subspace in which the boundary region is embedded. Samples away from the boundary (V2, etc.) tend to have a greater magnitude. These large values can have a considerable impact on the scattering matrix, distorting the information on the boundary structure. Therefore, it seems appropriate to emphasize how to move samples away from the border.

Chapter 6

Implementation and experiment result

In order to demonstrate and evaluate the proposed NDA approach, we have run several experiments on NIST-SRE 2010 benchmark datasets. Further, we have also provided a detailed comparison between our approach and several state of the art methods. Cross-validation method was used for refining the parameter values in each experiment.

A Gaussian PLDA system for the SRE10 extended eval was run using i-vectors from a gender-dependent 2048-UBM with 60-MFCCs. The i-vector extractor is also gender-dependent. It returns two cell arrays (target and non-target scores) for the 9 conditions.

The dataset used here in the experiment is collected from NIST-SRE10 plan [16], of which female development dataset has 18345 total i-vectors and is labelled into 1880 speakers. Male development dataset has 13057

i-vectors which represent total speech files, those i-vectors are labelled by 1281 speakers. Both T matrix in i-vector system and Phi matrix in PLDA system are pre-trained in advanced using the development data, verification was done using testing dataset. Both training and evaluation process are done in gender dependent phase, that is, male and female separately.

In both male and female system, i-vectors are regarded as front-end features produced by a generative model which represents one speaker's speech segment. I-vectos has 400 dimension, speaker-specific subspace dimension is set to 150, which refers to the dimension of hidden variable beta in PLDA system. Features are pre-processed by NDA reduction to 300 dimension, with parameter K set to 15. Length normalization is applied to all i-vectors. The two steps of the length normalization are whitening and then projection into unit sphere and the noise term has full-covariance.

The results for SRE10 extended 1conv-1conv is evaluated in terms of EERs and DCFs [16], which are calculated by Bosaris toolkit [17] from PLDA scores.

As shown in the table 6.1, 9 DET shows 9 conditions indicated in NIST-SRE10 plan, whose target and non target trials present in the following. First two columns present the result of original i-vector G-PLDA system without any dimension reduction process, while the other two sections indicate the implementation with LDA and NDA, respectively. It is clear to observe with LDA, EER remains the similarity with original results while minDCF has huge drop. While in NDA, Equal error rate has consistently reduction with respect to both original and LDA based system, and it also retains the good

Table 6.1: Comparison between no pre-process, LDA and NDA in i-vector PLDA system, in terms of EER and minDCF

			no pre-process		LDA		NDA	
DET	TGT	NTGT	EER	newDCF	EER	newDCF	EER	newDCF
1	4304	795995	1.57%	0.2479	1.55%	0.0154	1.41%	0.0134
2	15084	2789534	2.55%	0.4735	2.45%	0.0242	2.22%	0.0222
3	3989	637850	2.47%	0.4584	2.48%	0.0247	2.28%	0.0222
4	3637	756775	1.73%	0.3380	1.78%	0.0177	1.51%	0.0150
5	7169	408950	1.85%	0.3629	1.86%	0.0185	1.50%	0.0150
6	4137	461438	4.25%	0.7488	4.30%	0.0430	3.47%	0.0342
7	359	82551	5.64%	0.6825	5.40%	0.0539	4.10%	0.0393
8	3821	404848	1.72%	0.4162	1.87%	0.0183	1.55%	0.0149
9	290	70500	1.10%	0.1801	0.93%	0.0080	0.89%	0.0083

result of minDCF reduction. Both LDA and NDA has channel compensation feature which reflects in the minDCF decrease, and NDA does show good and consistent discriminant feature compared with LDA.

Chapter 7

Conclusions

In this thesis, an overview of speaker recognition system is presented, which contains from feature extraction, modelling technique to the choose of classier. From traditional method to state-of-the-art system. The impact of pre-process of i-vectors is investigated, we emphasize the effective of dimension reduction algorithm i.e. LDA. The improvement here is made by the introduction of NDA, which modifies the calculation of between scatter matrix and release the drawback of traditional fisher LDA. Satisfied result has been shown by the implementation of NDA in NIST 2010 SRE. The research effort to tackle the problem for speaker verification context has been significantly increased in recent years. Further more, research directions currently and in the future are as follows: Deep Neural Network (DNN), Metric Learning Technique, Sparse methods, Dimensionality reduction techniques, Miscellaneous Opportunities. Huge effort and attention should be effort into ASV research and we wish a better future will come soon.

Bibliography

[1] *"A Tutorial on Text-Independent Speaker Verification"* F Bimbot - 2004

[2] *"Speaker Recognition: A Tutorial"* JOSEPH P. CAMPBELL, JR., SENIOR MEMBER, IEEE

[3] *An overview of text-independent speaker recognition: From features to supervectors* Tomi Kinnunen , Haizhou Li - 2009

[4] *"Speaker adaptation using constrained estimation of Gaussian mixtures"* V. Digalakis, D. Rtischev, and L. Neumeyer, IEEE Trans. Speech Audio Process., vol. 3, no. 5, pp. 357366, September 1995.

[5] *"Speaker Verification Using Adapted Gaussian Mixture Models"* Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn

[6] *"Support Vector Machines Using GMM Supervectors for Speaker Verification"* W. M. Campbell, Member, IEEE, D. E. Sturim, Member, IEEE, and D. A. Reynolds, Senior Member, IEEE

[7] *"Front-end factor analysis for speaker verification"* N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, IEEE Trans. Audio Speech Lang. Process., vol. 19, no. 4, pp. 788798, 2011.

[8] *"Probabilistic Linear Discriminant Analysis for Inferences About Identity"* Simon J.D. Prince, James H. Elder

[9] *"Analysis of I-vector Length Normalization in Speaker Recognition Systems"* Daniel Garcia-Romero and Carol Y. Espy-Wilson in Proc. INTERSPEECH, Florence, Italy, August 2011, pp. 249252.

[10] *"Deep neural networks for extracting baum-welch statistics for speaker recognition"* P Kenny

[11] *"Deep neural networks for acoustic modeling in speech recognition"* Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury

[12] *"Score Normalization for Text-Independent Speaker Verification Systems"* Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas

[13] *"Introduction to Statistical Pattern Recognition 2nd Ed"* Keinosuke Fukunaga. New York: Academic press, 1990.

[14] *"Nonparametric Discriminant Analysis"* K. FUKUNAGA AND J. M. MANTOCK

[15] *"The IBM 2016 Speaker Recognition System"* Seyed Omid Sadjadi, Sriram Ganapathy, Jason W. Pelecanos IBM Research, Yorktown Heights, NY, USA Dept. of Electrical Eng., Indian Institute of Science, Bangalore, India

[16] *"The NIST Year 2010 Speaker Recognition Evaluation Plan"*

[17] *"https://sites.google.com/site/bosaristoolkit/"*