# 國 立 交 通 大 學

電機與控制工程學系

碩士論文

基於獨立成分分析(ICA)與廣義機率遞減法則(GPD)之

語者辨識與確認技術

A Text-Independent Speaker Verification Technique Based on

ICA and GPD Methods for Imposter-Rejection

研 究 生：陳晴慧

指導教授：林進燈 博士

中華民國 九十四 年 七 月

# 基於獨立成分分析(ICA)與廣義機率遞減法則(GPD)之語者辨識與確認技術

# A Text-Independent Speaker Verification Technique Based on ICA and GPD Methods for Imposter-Rejection


研 究 生：陳晴慧　　Student：Ching-Hui Chen

指導教授：林進燈 博士　　Advisor：Dr. Chin-Teng Lin

國立交通大學

電機與控制工程學系


碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國 九十四 年 七 月

# 基於獨立成分分析(ICA)與廣義機率遞減法則(GPD)之語者辨識與確認技術

學生：陳晴慧　　指導教授：林進燈 博士

國立交通大學電機與控制工程研究所

## 摘 要

本論文提出一個新的詞語不相關的語者辨識系統。使用常見的獨立成分分析法（Independent Component Analysis, ICA），找出原始特徵－梅爾倒頻譜參數（Mel-Frequency Cepstral Coefficient, MFCC）中蘊含重要資訊且互相獨立的成分，並將這些獨立成分用於特徵轉換上。此外，運用找出來的 ICA 基底進行降維的動作。因此，在所提出的辨識系統中，以 ICA 基底所轉換出來的特徵做為替代 MFCC 的新特徵。由實驗結果可以證明，使用新特徵的辨識結果比 MFCC 的辨識結果佳。而分類器方面，應用廣義機率遞減法則（General Probability Descent, GPD）對高斯混合模型辨識器（Gaussian Mixture Model, GMM）做最佳化的動作，以取代傳統上使用的最大相似度法則（Maximization Likelihood, ML）。由於 GPD 的目標是直接對辨識錯誤率做最小化的動作，因此 GPD 將決策規則（decision rule）以函數的形態納入整體架構中，故 GPD 適合用來最佳化辨識模型的參數。在以 GMM 為主體的系統中，本論文將具體實現 GPD 法則。經實驗證明，和傳統以 MFCC 及 ML 做為最佳化 GMM 的架構相比，本論文所提出的新架構有較佳的辨識結果。

# A Text-Independent Speaker Verification Technique Based on ICA and GPD Methods for Imposter-Rejection

Student: Ching-Hui Chen Advisor: Dr. Chin-Teng Lin

Institute of Electrical and Control Engineering

National Chiao-Tung University

## Abstract

In this thesis, we propose a novel text-independent speaker recognition system. A decomposition called the independent component analysis (ICA) is used to find out the most important and independent components of the original feature MFCC for the process of feature transformation. We also can reduce the dimension of the features depending on the ICA basis. These ICA-based features are used as our new features in the proposed system. The experiments have shown that using new features has an improvement on using MFCC. In addition, in the classifier phase, we apply the general probability descent (GPD) method to optimize the GMM recognizer instead of the conventional method such as the maximization likelihood (ML) method. That's because the objective of GPD is to minimize the recognition error rate directly. The decision rule of GPD appears in a function form in the overall criterion and it is suitable for the model parameter optimization. We present an implementation of the GPD method in a GMM-based system. The experiments have shown that the recognition rate of our proposed system is higher than the rate of the system with MFCC as the features and the ML-based GMM as the recognizer. It means that the experimental results verify our proposed system with ICA-based features and the GPD-based GMM recognizer.

# 誌　謝

本論文能順利完成，首先要感謝指導教授林進燈博士這兩年來的指導與提攜，讓我學習到許多寶貴的經驗，在研究方法上亦獲益良多。也要感謝口試委員們的建議與指教，使得本論文更為完整。

其次，要感謝的是實驗室的劉得正學長，在語音知識、程式技巧以及論文的撰寫上，學長給了我許多的建議與鼓勵，讓我能很快的瞭解這個研究領域，衷心感謝學長的一切幫助。

此外，也要謝謝實驗室裡眾多的學長、同學與學弟們，在日常生活中的協助與陪伴，讓研究的日子充實也充滿歡笑。

最後感謝家人對我的栽培與支持，讓我能安心地完成我的學業。

# Contents

**Chinese Abstract** ..... I
**Abstract** ..... II
**Acknowledgement** ..... III
**List of Figures** ..... VI
**List of Tables** ..... VII
**Chapter 1 Introduction** ..... 1
1.1 Motivation ..... 1
1.2 Literature Survey ..... 2
1.3 Organization of Thesis ..... 5
**Chapter 2 Framework of Independent Component Analysis and General Probability Descent Method used in the Speaker Recognition System** ..... 6
2.1 Introduction ..... 6
2.2 Independent Component Analysis ..... 6
2.2.1 Policy of ICA ..... 6
2.2.2 Implementation of FastICA Algorithm ..... 10
2.3 General Probability Decent Method ..... 14
2.3.1 Discriminative Function Approach (DFA) ..... 14
2.3.2 Generalized Probabilistic Descent (GPD) Method ..... 15
2.3.3 Summarize Advantages of GPD Formalization ..... 21
**Chapter 3 Speaker Recognition System Based on ICA and GPD Optimizer** ..... 23
3.1 Overall Speaker Recognition System ..... 23
3.2 Each Block of Speaker Recognition System ..... 24
3.2.1 Feature Extraction ..... 25
3.2.2 ICA Algorithm ..... 26
3.2.3 GPD-Based GMM ..... 27
**Chapter 4 Experiment Results and Discussion** ..... 32
4.1 Introduction ..... 32
4.2 Experiment Database ..... 33

4.3 Experiment Result ..... 33
4.4 Discussion ..... 42
**Chapter 5 Conclusion and Future Work** ..... 43
5.1 Conclusion ..... 43
5.2 Future Work ..... 44
**Bibliography** ..... 46

# List of Figures

Fig. 1.1 Speaker Recognition System ..... 2

Fig. 2-1 Both the pdf of Super-Gaussian and Sub-Gaussian ..... 8

Fig. 2-2 Block diagram of the FastICA algorithm ..... 13

Fig. 3-1 Training phase of our speaker recognition system for each speaker *s*. ..... 24

Fig. 3-2 Test phase of our speaker recognition system ..... 24

Fig. 3-3 Block diagram of Feature Extraction ..... 26

Fig. 3-4 Block diagram of GPD-based GMM model for the training phase. ..... 30

Fig. 3-5 Block diagram of GPD-based GMM model for the test phase. ..... 31

Fig. 4-1 Sketch of the feature and the GMM model of Experiment I ..... 34

Fig. 4-2 Sketch of the feature and the GMM model of experiment II ..... 35

Fig. 4-3 Sketch of the feature and the GMM model of experiment III ..... 36

Fig. 4-4 Sketch of the feature and the GMM model of experiment IV ..... 37

Fig. 4-5 the recognition rates of four experiments ..... 38

Fig. 4-6 the error true rates of four experiments ..... 38

Fig. 4-7 the error false rates of four experiments ..... 39

Fig. 4-8 the recognition rates of four experiments ..... 39

Fig. 4-9 the error true rates of four experiments ..... 40

Fig. 4-10 the error false rates of four experiments ..... 40

Fig. 4-11 the recognition rates of four experiments ..... 41

Fig. 4-12 the error true rates of four experiments ..... 41

Fig. 4-13 the error false rates of four experiments ..... 42

# List of Tables

Table 4-1 Recognition Results of Experiment I for 5 customers (71 imposters).....34

Table 4-2 Recognition Results of Experiment I for 10 customers (66 imposters).....34

Table 4-3 Recognition Results of Experiment I for 20 customers (56 imposters).....34

Table 4-4 Recognition Results of Experiment II for 5 customers (71 imposters) .....35

Table 4-5 Recognition Results of Experiment II for 10 customers (66 imposters) .....35

Table 4-6 Recognition Results of Experiment II for 20 customers (56 imposters) .....35

Table 4-7 Recognition Results of Experiment III for 5 customers (71 imposters).....36

Table 4-8 Recognition Results of Experiment III for 10 customers (66 imposters)....36

Table 4-9 Recognition Results of Experiment III for 20 customers (56 imposters)....36

Table 4-10 Recognition Results of Experiment IV for 5 customers (71 imposters)....37

Table 4-11 Recognition Results of Experiment IV for 10 customers (66 imposters)..37

Table 4-12 Recognition Results of Experiment IV for 20 customers (56 imposters)..37

# Chapter 1
# Introduction

## 1.1 Motivation

Recently, there has been a noticeable research in the use of biometrics characteristics as a means of recognizing a person such as human voice, fingerprint, iris structure, facial characteristics and so on. Among the above characteristics, the speaker recognition system is the most convenient way to the user because one does not have to raise his/her hand nor move to the sensor. What the user needs to do is just opening his/her mouth and then saying some word. Especially in text-independent speaker recognition, the user can say anything he/she wants. Speaker recognition [1],[2] is generally classified into two major categories, i.e. speaker identification and speaker verification. The task of the former is to identify an unknown speaker from a known population based on the individual's utterances. On the other hand, speaker verification is the process of verifying the identity of a claimed speaker from a known population. The interest of this thesis focuses on the text-independent speaker identification to determine which one the speaker is, and speaker verification to judge a speaker as a customer or an impostor, i.e. if the speaker is not a customer, we will reject his/her claim. A common speaker recognition system is shown in Fig. 1. First, the features are extracted from the speech signal and then they will be used as inputs to a classifier. Second, the classifier makes the final decision regarding identification or verification.

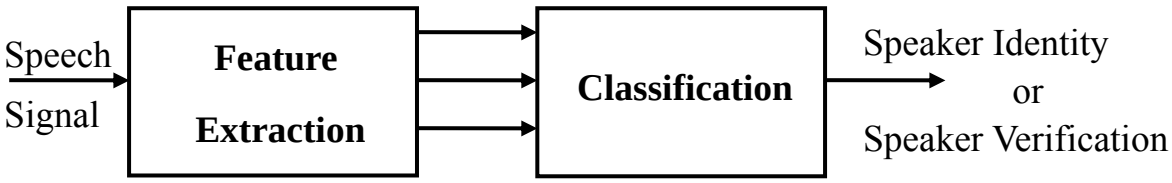



Fig. 1.1 Speaker Recognition System

Speaker recognition is expected to create new services such as the entrance guard system, phone banking, security control for confidential information areas, and remote access to computers. However, the current performance of state-of-the-art speaker recognition system is substantially inferior to the human performance. For the safety purpose, we must enhance the speaker recognition performance, which means we have to raise the recognition rate of the system. Therefore, the major objective of this thesis is to improve the performance of the speaker recognition system.

## 1.2 Literature Survey

When we obtain a speech signal, we will not use them directly to recognize a speaker because of its huge computation and messy representation. Hence we must extract the features hidden in the speech signal. So feature extraction is the essential process in speaker recognition systems. The popular and useful feature extraction approaches focus on the spectrum of the speech signals, and most of the current proposed speaker recognition systems use either the mel-frequency cepstral coefficients (MFCCs) or the linear predictive cepstral coefficients (LPCCs) as feature vectors. MFCCs are calculated based on the energy accumulated in the frequency filter banks whose ranges are decided according to the mel-scale [3]; while LPCCs is depended on the linear predictive coding.

Further, when we extract the feature, some useful modification can be done. For example, we can apply the independent component analysis (ICA) for extracting an optimal basis to the problem of finding efficient features for a speaker because ICA has been shown a highly effective in extracting the features from the given set of observed speech signals [4]-[6]. By using ICA, we can detect independent components of the MFCC features, but we may guess some independent components of all should be more important than the others. Therefore, we could only choose some components to achieve dimension reduction and computation saving.

After extracting features, a speaker model which represents each speaker in the speaker recognition system will be built in the training phase and then be used for speaker matching in the test phase. The modeling approaches are various, including the artificial neural network (ANN) [7], the vector quantization (VQ) [8],[9], the Gaussian mixture models (GMMs) [10],[11], the hidden Markov model (HMM) [12]-[14] and so on. In 1995, Reynolds demonstrated that the GMM-based classifier works well in text-independent speaker recognition even with speech features that contain rich linguistic information like MFCCs [15]. GMM provides a probability model of the underlying sounds of a speaker's voice. It uses several Gaussian density functions to model a speaker and each density function has its own mean and variance. For a feature vector denoted as $x_j$, the mixture density for one speaker is defined as $p(x_j \mid \lambda_s) = \sum_{i=1}^{M} p_i^s b_i^s(x_j)$. The density is a weighted linear combination of $M$ component uni-modal Gaussian densities $b_i^s(x_j)$, each parameterized by a mean vector $\vec{\mu}_i^s$ and covariance matrix $\Sigma_i^s$. Collectively, the parameters of a speaker's density model are denoted as $\lambda_s = \{p_i^s, \vec{\mu}_i^s, \Sigma_i^s\}$ and maximum likelihood (ML) estimates of the model parameters are obtained by using the expectation maximization

(EM) algorithm. Therefore, for an utterance $X = \{x_1, ..., x_N\}$ and a reference group of speakers $\{s_1, s_2, ..., s_S\}$ represented by models $(\lambda_1, \lambda_2, ..., \lambda_S)$, the identification is executed by the maximum likelihood classification rule $\hat{s} = \arg\max_{1 \le s \le S} p(X \mid \lambda_S)$ which decides who the candidate speaker [16] is.

Although GMM-based classifier works well in text-independent speaker recognition as mentioned above, there is one vital drawback of the model. That is the estimation error, either in parameter or in distribution, which does not immediately translate into the recognition performance of the recognizer that uses the estimated distributions [17]. An alternative approach is to directly design the recognizer to minimize the recognition error rate, so as to allow optimization of the recognizer parameters. This kind of approach is often called discriminative training. Therefore we add a minimum recognition error formulation and a generalized probabilistic descent (GPD) algorithm to form a foundation for the discriminative training approach. Another advantage of using the GPD method is that the structure of the conventional speech recognizer can be kept intact without modification. This can be convenient for the designer to implement the algorithm. The details of GPD will be described in Chapter 2.

In the following, we will describe the framework of our proposed speaker recognition system briefly.

First, we choose the MFCCs as our feature since the mel-scale mimics the human hearing which is sensitive to the sound in low-frequency domain. After the feature of each frame has been extracted, we use ICA to convert the original feature representation by MFCC into a new one by finding out the independent component of the feature. And then, we use the GPD-based GMMs to construct a model for each speaker. The GPD method is used to enhance the GMM model for considering the overall recognition system and reducing the overall system error. The detail of the

speaker recognition system will be described in Chapter 3.

## 1.3 Organization of Thesis

This thesis is organized as follows: Chapter 2 reviews the ICA algorithm and the GPD method. Chapter 3 describes the proposed structure of the speaker recognition, including MFCCs, the ICA features, and the GMM model with GPD. Chapter 4 depicts the used database and shows the experimental results to verify the performance of our speaker recognition system as mentioned in Chapter 3. The conclusions of this thesis and the future work are given in Chapter 5.

# Chapter 2

# Framework of Independent Component Analysis and General Probability Descent Method used in the Speaker Recognition System

## 2.1 Introduction

As mentioned above, the ICA is used to find out the most important and independent components of the MFCCs features, and the GPD method is taken to consider the whole situation for reducing the overall system error. Both of them are the main parts of this thesis, and therefore we will introduce them more detailed in this chapter.

## 2.2 Independent Component Analysis

In this subsection, we will show the policy of how to find independent components from the input vectors. And next, a technique called FastICA will be described to find the independent components.

### 2.2.1 Policy of ICA

Assume that the input vector **x** is distributed according to the ICA data model and **s** is the independent components:

$$x = As, \tag{2.1}$$

where $A$ is the mixing matrix. For simplicity, we also assume that all the

independent components have identical distributions and the unknown mixing matrix $A$ is square. After estimating matrix $A$, we can obtain the independent component by:

$$s = Wx, \text{ where } W = A^{-1}. \tag{2.2}$$

Thus one of the independent components can be considered as a linear combination of $x_i$, denoted by:

$$y = w^T x = \sum_i w_i x_i, \tag{2.3}$$

where $w$ is some weight vector to be determined. If $w$ were one of the rows of the inverse of $A$, then $y$ would actually be one of the independent components. However, we cannot determine such a $w$ in practice because of no knowledge of matrix $A$.

In order to find an estimator that gives a good approximation of $w$, let us redefine the variables as:

$$z = A^T w. \tag{2.4}$$

Then we get

$$y = w^T x = w^T As = z^T s. \tag{2.5}$$

Obviously, $y$ is a linear combination of $s_i$, with weights $z_i$. Since a sum of any two independent random variables is more Gaussian than the original variables, $z^T s$ is more Gaussian than any of $s_i$. In addition, $s_i$ was assumed to have identical distributions, so only one of the elements $z_i$ of $z$ is nonzero. Therefore, we could take $w$ as a vector that maximizes the non-gaussianity of $w^T x$. That means we need to find all these local maxima in order to find several independent components.

For simplify computation, we assume that $y$ is centered (zero-mean) and has variance equal one. The classical measure of nongaussianity is kurtosis, defined by:

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 = E\{y^4\} - 3. \tag{2.6}$$

Thus, for any Gaussian variable, its kurtosis is zero, and on the other hand, kurtosis is nonzero for most non-gaussian random variables. Besides, we call the random variables with a positive kurtosis as the super-Gaussian, and those with a negative kurtosis as the sub-Gaussian. Super-Gaussian random variables have typically a spiky pdf with heavy tails; on the other hand, sub-Gaussian random variables have a plat pdf. They are illustrated in Fig. 2-1.

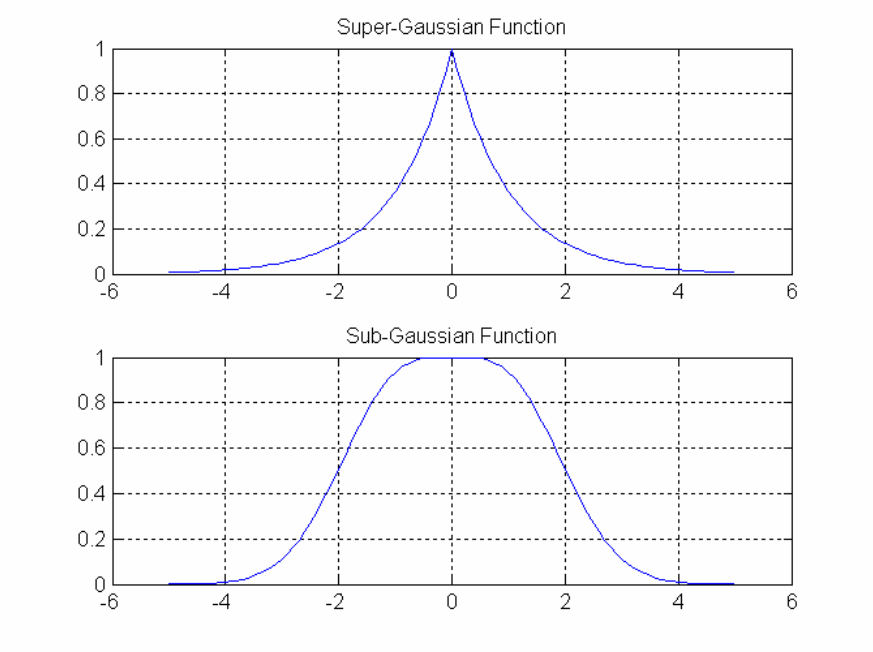



Fig. 2-1 Both the pdf of Super-Gaussian and Sub-Gaussian

Therefore we can use the absolute value or the square of kurtosis to measure non-gaussianity in ICA.

In practice, we could start from some weight vector $w$ and use a gradient method or one of their extensions for finding a new $w$. However, there are some drawbacks in the kurtosis method and the main problem is that kurtosis is very sensitive to outliers. It means that kurtosis is not a robust measure of non-gaussianity. For this reason, we seek for other measures of non-gaussianity. Another important

measure is given by negentropy which is based on the information-theoretic quantity of entropy. Entropy is closely related to the coding length of the random variable.

Let us define entropy $H$ for a discrete random variable $Y$ as:

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i) . \tag{2.7}$$

If the random variables and vectors are continuous, we call this kind of entropy as differential entropy. Differential entropy $H$ of a random vector $y$ with density $f(y)$ is defined as:

$$H(y) = -\int f(y) \log f(y) dy . \tag{2.7}$$

Because a Gaussian variable has the largest entropy among all random variables of equal variance [18], we can use entropy as a measure of non-gaussianity. In order to obtain a measure of non-gaussianity that is zero for a Gaussian variable and always nonnegative, a slightly modified version of the definition of differential entropy, called negentropy, is redefined as:

$$J(y) = H(y_{gauss}) - H(y) , \tag{2.8}$$

where $y_{gauss}$ is a Gaussian random variable of the same covariance matrix as $y$. As eq.(2.8) mentioned, negentropy is always non-negative and it is zero only when $y$ has a Gaussian distribution. The drawback of negentropy is its difficult computation and then simpler approximations of negentropy will be discussed next.

The classical method of approximating negentropy is using higher-order moments as follows:

$$J(y) \approx \frac{1}{12} E\left\{y^3\right\}^2 + \frac{1}{48} kurt\left(y^2\right). \tag{2.9}$$

However, the validity of such approximations may be rather limited. These approximations will suffer from the non-robustness encountered with kurtosis. Therefore new approximations were developed based on the maximum-entropy

principle [19]. The approximation is showed below:

$$J(y) \approx \sum_{i=1}^{p} k_i \left[ E\{G_i(y)\} - E\{G_i(\nu)\} \right]^2 , \tag{2.10}$$

where $k_i$ are some positive constants, $\nu$ is a Gaussian variable of zero mean and unit variance, the variable $y$ is assumed to be of zero mean and unit variance, and the functions $G_i$ are some non-quadratic functions. In this case, we use only one non-quadratic function $G$, and then the approximation becomes:

$$J(y) \propto \left[ E\{G(y)\} - E\{G(\nu)\} \right]^2 . \tag{2.11}$$

If $y$ is symmetric, eq. (2.11) is a generalization of the moment-based approximation in eq. (2.9). In particular, choosing $G$ that does not grow too fast, one can obtain more robust estimators.

Thus, we obtain approximations of negentropy that give a very good compromise between the properties of the two classical non-gaussianity measures given by kurtosis and negentropy. They are conceptually simple, fast to compute, yet have appealing statistical properties, especially robustness.

### 2.2.2 Implementation of FastICA Algorithm

We have introduced the measures of non-gaussianity, which are the objective functions for ICA estimation. In practice, we also need an algorithm for maximizing the contrast function such as eq. (2.11). Then, we introduce a very efficient method of maximization suited for this task and how to find the ICA basis in the following. We will first show the one-unit version of FastICA and extend it to the several-unit version.

## FastICA for one unit

Here, the 'unit' means a computational unit, which is an artificial neuron, having a weight vector $w$ that is able to be updated by a learning rule. The learning rule of FastICA finds a direction for $w$ such that the projection $w^T x$ maximizes non-gaussianity measured by eq. (2.11). Recall that the variance of $w^T x$ must be constrained to unity, which is equivalent to constraining the norm of $w$ to be unity for whitened data

Because the non-quadratic function $G$ used in eq. (2.11) must not grow too fast for obtaining a robust estimator, we choose $G$ as [4]:

$$G(y) = -y \cdot \exp(-ay^2/2), \tag{2.12}$$

and the derivative of $G$ is:

$$g(y) = (ay^2 - 1) \cdot \exp(-ay^2/2), \tag{2.13}$$

where $1 \leqq a \leqq 2$ is some suitable constant.

The basic form of the FastICA algorithm is shown below:

1. Center the data to make its mean zero.
2. Whiten the data to give $p$.
3. Choose an initial weight vector $w$ of unit norm.
4. Let $w \leftarrow E\{pG(w^T p)\} - E\{g(w^T p)\}w$, where $G$ and $g$ is defined in eq(2.12) and eq(2.13).
5. Let $w \leftarrow w/\|w\|$.
6. If not converged, go back to step 4.

## FastICA for several units

The one-unit FastICA algorithm estimates only one of the independent components or one projection pursuit direction. To estimate several independent components, we run the one-unit FastICA algorithm using several units with weight vectors $w_1,...,w_n$. To prevent different vectors from converging to the same maxima, we decorrelate the outputs $w_1^T x,...,w_n^T x$ in every iteration.

A simple way of achieving decorrelation is a deflation scheme based on a Gram-Schmidt-like decorrelation. This means we must estimate the independent components one by one. When we have estimated $p$ independent components, or $p$ vectors $w_1,...,w_p$, we run the one-unit algorithm for $w_{p+1}$, and after every iteration step, subtract the projections $w_{p+1}^T w_j w_j$ from $w_{p+1}$, $j = 1...p$, and then renormalize $w_{p+1}$. The more detailed steps are listed below:

1. Choose $n$, the number of independent components to estimate. Set $p \leftarrow 1$.
2. Initialize $w_p$ randomly.
3. Do an iteration of a one-unit algorithm on $w_p$.
4. Do the following decorrelation:

$$w_{p+1} \leftarrow w_{p+1} - \sum_{j=1}^{p} w_{p+1}^T w_j w_j \tag{2.14}$$

5. Normalize $w_p$ by dividing it by its norm:

$$w_{p+1} \leftarrow w_{p+1} / \sqrt{w_{p+1}^T w_{p+1}} \tag{2.15}$$

6. If $w_p$ has not converged, go back to step 3.
7. Set $p \leftarrow p+1$. If $p$ is not greater than the desired number of independent components, go back to step 2.

When the algorithm stops, we will obtain the independent components of the original features MFCCs. Figure 2-2 shows the flowchart of the FastICA algorithm.

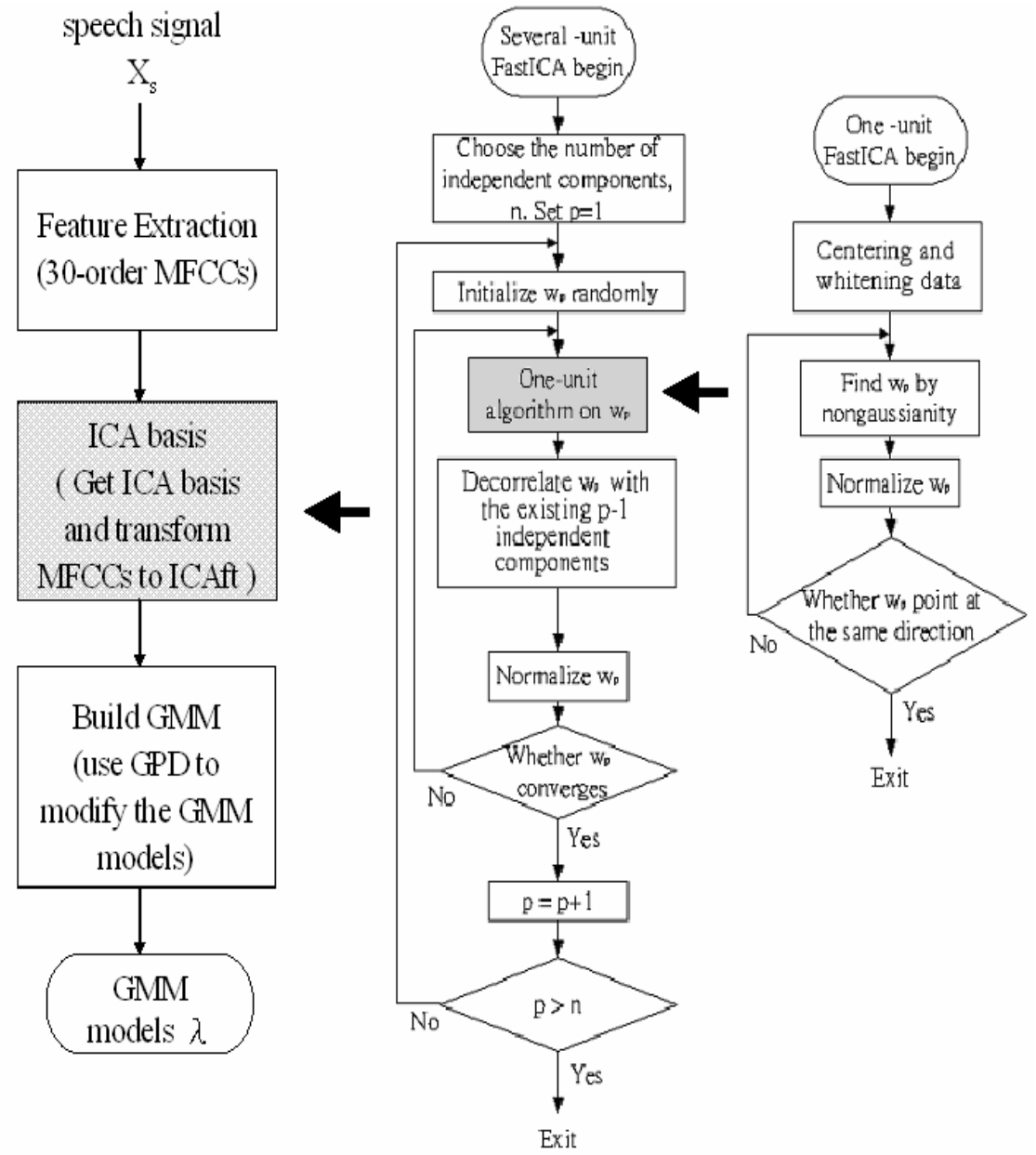



Fig. 2-2 Block diagram of the FastICA algorithm

## 2.3 General Probability Decent Method

Traditionally, the classifiers of most existing recognizers have been designed based on the design principle of the maximum likelihood (ML) algorithm; that is the expectation-maximization (EM) method, which is an extended ML estimation method for incomplete data [21], and segmental k-means clustering [22] are used for training the acoustic model. However, the conventional ML-based approach has a basic problem in which the function form of the class distribution (the conditional probability density) function to be estimated is rarely known in practice and the likelihood maximization of these estimated functions is not direct with regard to the minimization of classification errors. Besides, the ML-based approach covers only the classifier design; it does not optimize the overall system [23].

One of the solutions for solving the above problem and meeting the need of improvement in the recognition performance is the generalized probabilistic descent (GPD) method, which is based on a discriminative function approach (DFA), developed for classifier design [20]. The GPD algorithm was shown to be consistent with the objective of minimizing the classification error rate and to be very useful in various pattern recognition tasks. This thesis is therefore devoted to providing the GPD approach to the speaker recognition in a GMM-based system.

### 2.3.1 Discriminative Function Approach (DFA)

Consider a set of training samples $X = \{x_1, x_2, ..., x_N\}$, where each $x_j$ is a D-dimensional vector and is known to belong to one of $S$ classes $C_s, s = 1, 2, ..., S$. A classifier comprises a set of parameters and a decision rule.

In DFA, a discriminant function $g_s(x_j; \lambda_s)$ is introduced for $C_s$ to measure

the class membership of the input $x_j$, where $\lambda_s$ is the parameters of classes $C_s$. The discriminant function can be a probability function, distance, similarity, or any reasonable type of measure. And then, use the discriminant function to implement the decision rule as shown below [23]:

$$C(x_j) = C_k, \ iff \ \ k = \arg \max_s g_s(x_j; \lambda_s), \tag{2.16}$$

This approach is more direct with regard to the minimization of classification errors than the ML-based approach where class model parameters are designed independently of each other. However, there is plenty of room left for improvement in the DFA, as summarized in the following:

1) Execution of rule (2.16) using an arbitrary measure as the discriminant function does not necessarily lead to the minimum error probability situation.
2) The design scope does not cover the overall recognizer.
3) Most of the existing training procedures are empirical or heuristic; that means their mathematical optimality is unclear.

### 2.3.2 Generalized Probabilistic Descent (GPD) Method

From the above reasons in subsection 2.3.1, GPD is motivated to design a novel method for pursuing the overall optimality of a recognizer.

The fundamental concept of the GPD formalization is directly used in the overall process of classifying a pattern $x_j$ in a smooth functional form that is suited for the use of a practical optimization method, especially gradient search optimization [22],[24]. In the following, we propose an embodiment of GPD for the GMM classifier in detail. GPD is formalized in the following three-step manner:

1) Choose GMM as a discriminant function

A Gaussian mixture density is a weighted sum of $M$ component densities, as shown in Fig. 2-3. $x_j$ is a D-dimensional vector, $b_i(x_j)$ are the component densities, and $w_i$ are the mixture weights, where $i = 1,...,M$ .

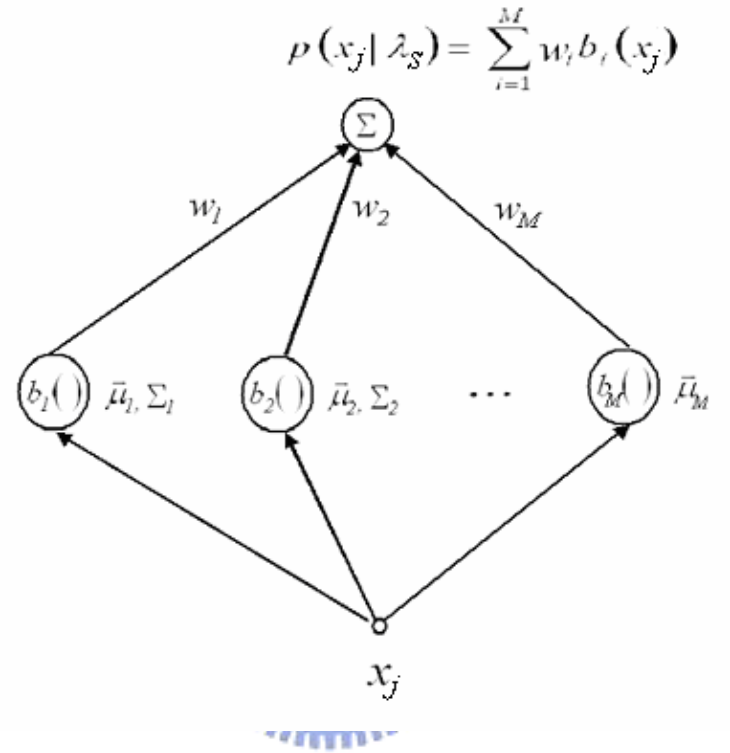



Fig. 2-3 Depiction of an $M$ Component Gaussian Mixture Density

Each component density is a $D$-variate Gaussian function of the form:

$$b_i\left(x_j\right) = \frac{1}{\left(2\pi\right)^{D/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(x_j - \vec{\mu}_i\right)' \Sigma_i^{-1}\left(x_j - \vec{\mu}_i\right)\right\} \qquad (2.17)$$

with mean vector $\vec{\mu}_i$ and covariance matrix $\Sigma_i$. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$. In order to simplify the following computation, we cite the Baum-Welch algorithm [25]. Based on an existing model $\lambda$, this algorithm transforms the objective function $p(x_j \mid \lambda)$ into a new function $Q(\lambda, \lambda')$ that essentially measures a divergence between the existing model $\lambda$

and an updated model $\lambda'$. It can be shown that $Q(\lambda,\lambda') \geq Q(\lambda,\lambda)$ implies $p(x_j \mid \lambda') \geq p(x_j \mid \lambda)$. Therefore, we define the discriminant function as:

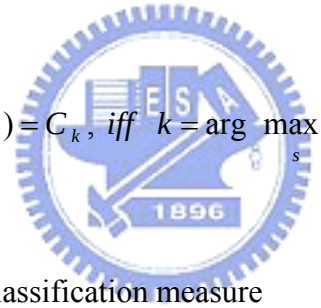
for classifier $s$,

$$g_s(x_j;\lambda_s) = Q_s(\lambda_s,\lambda_s') = \sum_{i=1}^{M} f_i(x_j)\log f_i'(x_j) + Z(1-\sum_{i=1}^{M} w_i) \quad (2.18)$$

where $f_i(x_j) = w_i b_i(x_j)$ and the symbols with a apostrophe means the updated data. The term $Z(1-\sum_{1}^{M} w_i)$ is used to make sure the sum of updated weights is one. Accordingly, the method achieves a smooth discriminant function for the pattern $x_j$.

Then, we use the discriminant function to implement the decision rule which is stated as eq. (2.16):

$$C(x_j) = C_k, \ iff \ \ k = \arg \max_s \ g_s(x_j;\lambda_s).$$



2) Define a smooth misclassification measure

The smooth optimization criterion is a function of the discriminant function $g_s(x_j;\lambda_s),\ s = 1,...,S$. Again, the classifier makes its decision for each pattern $x_j$ by choosing the largest of the discriminant function evaluated on $x_j$. The key to the new error criterion is to express the operation decision rule of (2.16) in a function form. Among many possibilities, the following is a typical definition of the class misclassification measure for $x_j\ (\in C_k)$:

$$d_k(x_j;\lambda_k) = -g_k(x_j;\lambda_k) + \left[\frac{1}{N-1}\sum_{n\neq k}^{N}\{g_n(x_j;\lambda_n)\}^{\mu}\right]^{1/\mu}, \quad (2.19)$$

where $\mu$ is a positive constant [26]. This misclassification measure is a

continuous function of the classifier parameter $\lambda$, and attempts to emulate the decision rule. A large $d_k(x_j;\lambda_k)$ implies that more definitely the input is misclassified. By varying the value of $\mu$, we can take all the competing classes into consideration in the process of optimizing the classifier parameter $\lambda$. To complete the definition of the objective criterion, the misclassification measure of (2.19) is used in the third step where the recognition error is counted.

3) Define the loss function

A general form of the loss function can be defined as:

$$l_k(x_j;\lambda_k) = l_k(d_k(x_j;\lambda_k)), \tag{2.20}$$

which is expressed as a function of the misclassification measure. The loss function $l$ is a sigmoid function. For minimum error classification, the following loss function is merely one of several possibilities:

$$l_k(d_k) = \frac{1}{1+\exp(-(\alpha d_k + \beta))}, \ (\alpha > 0) \tag{2.21}$$

with $\beta$ normally sets to zero and $\alpha$ sets to equal or greater than one. Apparently, when $d_k(x_j;\lambda_k)$ is much smaller than zero, which implies correct classification, virtually no loss is occurred. On the contrary, when $d_k(x_j;\lambda_k)$ is positive, it leads to a penalty which becomes a classification/recognition error count. That is, this formulation allows us to directly minimize the expected recognition error by gradient descent search methods.

This three-step method is suitable for classifier parameter optimization. Based on the criterion of (2.21), we use it to minimize the expected loss for the classifier parameter search.

## Optimization Method

There are various minimization algorithms which can be used to minimize the expected loss. Among them, the GPD method is a powerful algorithm that can accomplish this task. In the GPD-based minimization algorithm, the expected loss function $L(\lambda) = E[l_k(x_j;\lambda_k)]$ is minimized according to an iterative procedure.

We seek to minimize $L$ by adaptively adjusting λ in response to the incurred loss each time a training pattern $x_j$ is presented. The adjustment of $\lambda$ is according to:

$$\lambda_{t+1} = \lambda_t + \delta\lambda_t, \tag{2.22}$$

where $\lambda_t$ denotes the parameter set at the *t*-th iteration. The adjusted term $\delta\lambda_t$ is a function of the input pattern $x_j$ $(\in C_k)$ and the current parameter set $\lambda_t$, i.e., $\delta\lambda_t = \delta\lambda(x_j, \lambda_t)$. The magnitude of this term must be small such that the first-order approximation holds:

$$L(\lambda_{t+1}) \doteq L(\lambda_t) + \delta\lambda_t \nabla L(\lambda)|_{\lambda=\lambda_t}, \tag{2.23}$$

Then, we can obtain the equation:

$$E[L(\lambda_{t+1}) - L(\lambda_t)] = E[\delta L(\lambda_t)] = E[\delta\lambda(x_j, \lambda_t)]\nabla L(\lambda_t), \tag{2.24}$$

Therefore, the goal is to find an adaptation rule such that $E[\delta L(\lambda_t)] < 0$ and such that $\lambda_t$ converges to an at least locally optimum solution $\lambda^*$. The probabilistic descent algorithm is summarized in the following theorem.

**Probabilistic Descent Theorem** [27]:

Assume that a given pattern $x_j$ belongs to class $C_k$.

If the classifier parameter adjustment $\delta\lambda(x_j, \lambda_t)$ is specified by

$$\delta\lambda(x_j, \lambda_t) = -\varepsilon U \nabla l(x_j; \lambda_t), \tag{2.25}$$

where $\varepsilon$ is a small positive real number and $U$ is a positive-definite matrix which is often assumed for simplicity to be a unit matrix, then

$$E[\delta L(\lambda_t)] \le 0. \tag{2.26}$$

Furthermore, if an infinite sequence of randomly selected samples $x_j$ is used for learning and the adjustment rule of (2.25) is utilized with a corresponding learning weight sequence $\varepsilon(t)$ which satisfies

$$\sum_{t=1}^{\infty} \varepsilon(t) \to \infty, \tag{2.27}$$

$$\sum_{t=1}^{\infty} \varepsilon(t)^2 < \infty, \tag{2.28}$$

then the parameter sequence $\lambda(t)$ according to

$$\lambda(t+1) = \lambda(t) + \delta\lambda(x_j, \lambda(t)) \tag{2.29}$$

converges with probability one to $\lambda^*$ which is at least a local minimum of $L(\lambda)$.

It is obviously unrealistic to observe the infinitely repeated probabilistic descent adjustments. In practice, the learning coefficient $\varepsilon(t)$ is usually approximated by a finite monotonically decreasing function as

$$\varepsilon(t) = \varepsilon(0)\left(1 - \frac{t}{T}\right), \tag{2.30}$$

where $T$ is a preset number of adjustment repetitions.

The resulting adjustment rule using loss function (2.21) for the GMM parameters $\{w_i, \vec{\mu}_i, \Sigma_i\}$ are given as

$$w_i(t+1) = w_i(t) - \varepsilon(t)\nu_k\varphi_j\zeta_{j,i}, \tag{2.31}$$

$$\vec{\mu}_i(t+1) = \vec{\mu}_i(t) + \varepsilon(t)\nu_k\varphi_j\eta_{j,i}, \tag{2.32}$$

$$\Sigma_i(t+1) = (\Sigma_i(t)^{-1} - \varepsilon(t)\nu_k\varphi_j\rho_{j,i})^{-1}, \tag{2.33}$$

where

$$\nu_k = \alpha l_k(d_k)(1 - l_k(d_k)), \tag{2.34}$$

$$\zeta_{j,i} = \frac{f_i(x_j)}{w_i} - \frac{\sum_{i=1}^{M}(f_i(x_j)/w_i)}{M}, \text{ where } f_i(x_j) = w_i b_i(x_j), \tag{2.35}$$

$$\eta_{j,i} = -f_i(x_j)\Sigma_i^{-1}\left(x_j - \vec{\mu}_i\right), \tag{2.36}$$

$$\rho_{j,i} = \frac{1}{2} f_i(x_j)[\Sigma_i - \left(x_j - \vec{\mu}_i\right)\left(x_j - \vec{\mu}_i\right)'], \tag{2.37}$$

for $j = k$,

$$\varphi_j = -1, \tag{2.38}$$

for $j \neq k$,

$$\varphi_j = \frac{1}{N-1}\left[\frac{1}{N-1}\sum_{n\neq k}^{N}\left(\frac{g_n}{g_j}\right)^{\mu}\right]^{1/\mu - 1}, \tag{2.39}$$

In eq. (2.31)-(2.33), the adjustment is done for all of the patterns.

### 2.3.3 Summarize Advantages of GPD Formalization

The most important point of the GPD concept is to embed the entire process of a given recognition task into a smooth function. Therefore, we can optimize all of the adjustable system parameters in consistent with the design objective of minimizing recognition errors.

In addition, GPD has both mathematical rigor and a great degree of practicality. GPD was shown to provide attractive solutions to three of the four major DFA issues:

1) The design objective;

2) Optimization method;

3) Design consistency with unknown samples.

The forth DFA issue, which is the selection of the discriminant function form, has not been fully studied yet.

Because of the above advantages, we choose GPD to modify the GMM for speaker recognition.

# Chapter 3

# Speaker Recognition System Based on ICA and GPD Optimizer

## 3.1 Overall Speaker Recognition System

The framework of our speaker recognition system is shown in Fig. 3-1 and Fig. 3-2.

For the training phase, feature $MFCC_s$ is extracted from the original speech signal of speaker $s$, and then we use the FastICA algorithm to find the independent components of $MFCC_s$. Therefore, we transform $MFCC_s$ into feature $ICAft_s$ based on the basis found from the above step. In the next step, we use the $ICAft_s$ as the input of GMM to train the model. Among the structure, the GPD method is utilized to optimize the GMM recognizer. From the above steps, we could obtain the speaker recognition structure of each speaker $s$.

In the test phase of speaker recognition system, we also extract MFCC from the speech signal, and transform them by the ICA basis obtained in the training phase. Then, we use the new features to evaluate the degree (score) of matching the GMM model of some speaker. If the largest score, which is estimated from some model of speaker $k$, is smaller than a threshold we set in advance, then we will reject the speaker and take him/her as an imposter. Otherwise, we regard the speaker as one customer.

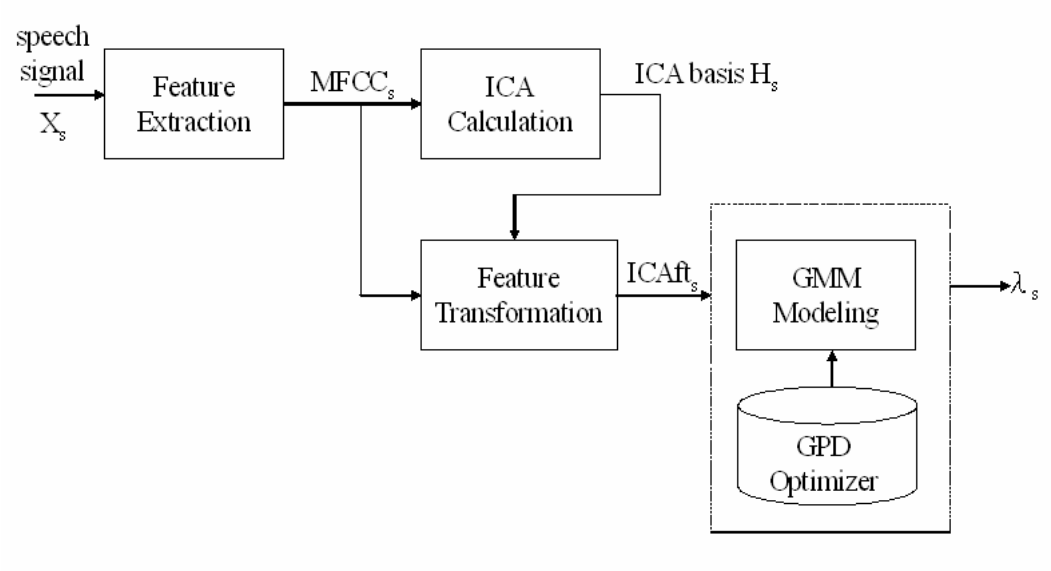



Fig. 3-1 Training phase of our speaker recognition system for each speaker *s*.

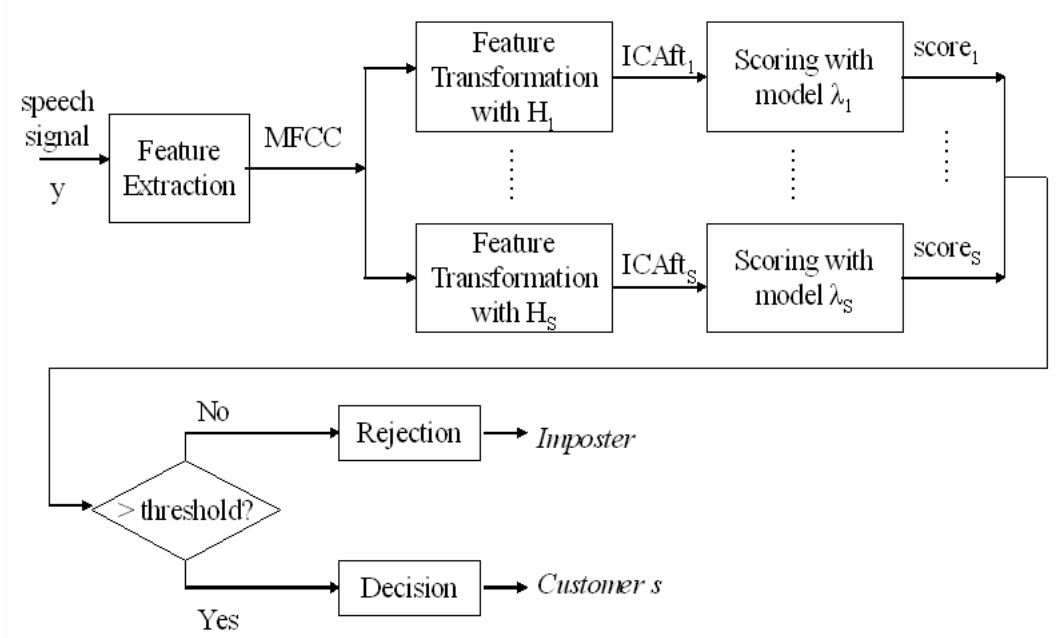



Fig. 3-2 Test phase of our speaker recognition system

## 3.2 Each Block of Speaker Recognition System

In this section, we will decompose the entire speaker recognition system into blocks. After that, we will detail each block of the recognition system.

### 3.2.1 Feature Extraction

MFCC is widely used in the automatic speech recognition (ASR) applications. It is primarily for the three reasons [28]: 1) The cepstral features are roughly orthogonal because of the DCT, 2) cepstral mean subtraction eliminates static channel noise, and 3) MFCC is less sensitive to additive noise than linear prediction cepstral coefficients (LPCC). The key component of MFCC responsible for noise robustness is the filter bank; the filters smooth the spectrum, reducing variation due to additive noise across the bandwidth of each filter.

First, the speech signal is pre-processed by a high-pass filter. Next, a segment (frame) of speech is windowed and transformed to the frequency domain via the fast Fourier transform (FFT) and then the magnitude spectrum of the utterance is passed through a bank of triangular-shaped filters whose center frequencies are spaced along the perceptually-motivated Mel frequency scale. Therefore, the energy output from each filter is log-compressed and transformed to the cepstral domain via the discrete cosine transform (DCT). The block of feature extraction is shown in Fig. 3-3.

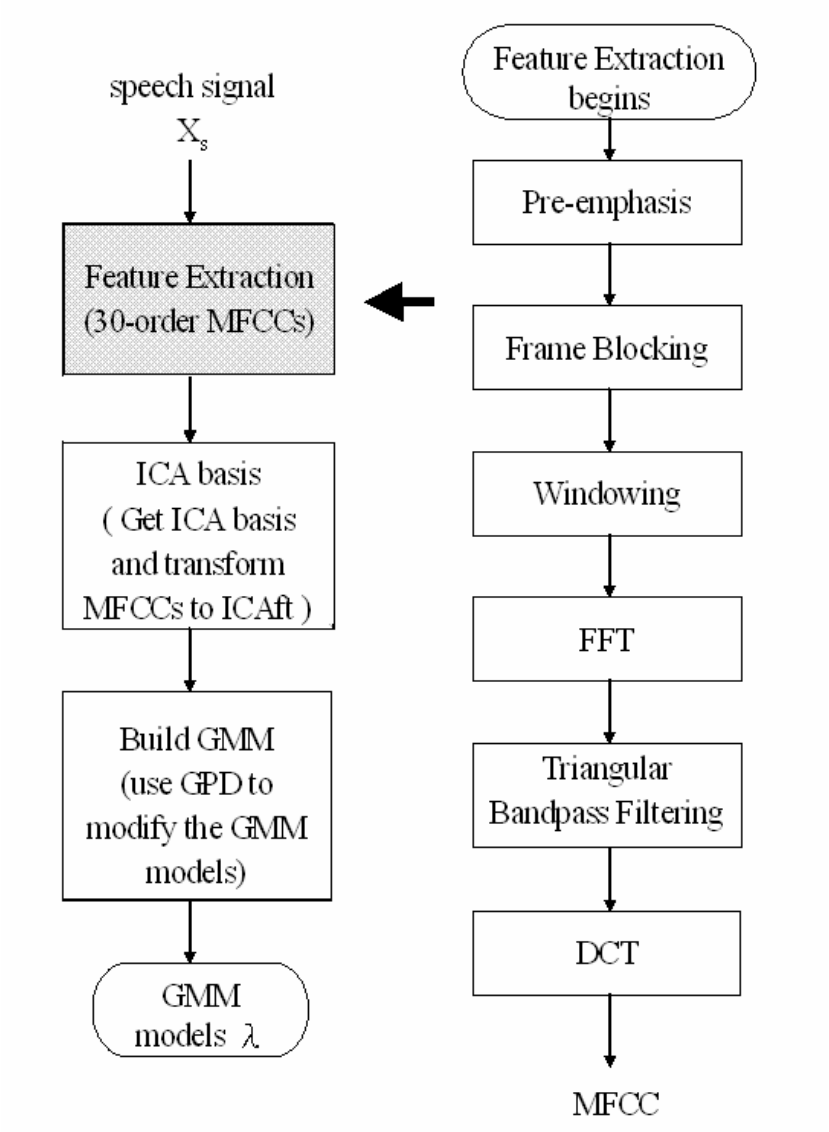



Fig. 3-3 Block diagram of Feature Extraction

### 3.2.2 ICA Algorithm

ICA can find a linear non-orthogonal coordinate system in multivariate data determined by high-order statistics. Its goal is to linearly transform the data such that the transformed variables are as statistically independent from each other as possible [29], [30]. Like data mining, ICA can extract the hidden predictive information from large databases and it is a powerful novel technology with great potential for finding the most important information in the data.

ICA not only decorrelates the signals but also reduces higher-order statistical

dependencies. We use it to find the most important and independent components of MFCC.

The block of ICA algorithm is shown in Fig. 2-2.

### 3.2.3 GPD-Based GMM

The most important concept of the GPD method is to formalize the overall procedure of the task into an optimized design process. Its objective is to directly minimize the recognition error rate.

One advantage of using GPD as the optimizer of the speaker recognition model is that the structure of the convention speaker recognizer can be kept intact without modification. This could demonstrate the practical value of the GPD method if it is to be incorporated in existing recognizer designs.

In addition, for reducing our computation, we will rewrite the equations in subsection 2.3.2. We assume that the covariance matrix is diagonal and the values of the elements in the diagonal are all the same for one Gaussian. That means, we can use a unique variance $V_i$ to replace the covariance matrix $\Sigma_i$. Then, eq. (2.17) is redefined as

$$b_i\left(x_j\right)=\frac{1}{\left(2\pi\right)^{D/2}\left|V_i\right|^{D/2}}\exp\left\{-\frac{1}{2V_i}\sum_{m=1}^{D}\left(x_{j,m}-\mu_{i,m}\right)^2\right\}. \qquad (3.1)$$

The definition of the discriminant function $g_s(x_j;\lambda_s)$, the classification decision rule, the class misclassification measure $d_k(x_j;\lambda_k)$, and the loss function $l_k(x_j;\lambda_k)$ are the same as eq.(2.16)-(2.21).

1) Discriminant Function:

For speaker $s$,

$$g_s(x_j;\lambda_s) = Q_s(\lambda_s,\lambda_s') = \sum_{i=1}^{M} f_i(x_j)\log f_i'(x_j) + Z(1-\sum_{i=1}^{M} w_i). \tag{3.2}$$

Decision Rule:

$$C(x_j) = C_k,\ iff\ \ k = \arg\ \max_s\ g_s(x_j;\lambda_s). \tag{3.3}$$

2) Misclassification Measure:

$$d_k(x_j;\lambda_k) = -g_k(x_j;\lambda_k) + \left[\frac{1}{N-1}\sum_{n\neq k}^{N}\{g_n(x_j;\lambda_n)\}^{\mu}\right]^{1/\mu}. \tag{3.4}$$

3) Loss Function:

$$l_k(x_j;\lambda_k) = l_k(d_k(x_j;\lambda_k)), \tag{3.5}$$

$$l_k(d_k) = (\,1+\exp(-(\alpha d_k+\beta))\,)^{-1},\ (\alpha>0). \tag{3.6}$$

The model adjustment is

$$\lambda(t+1) = \lambda(t) + \delta\lambda(x_j,\lambda(t)), \tag{3.7}$$

$$\delta\lambda(x_j,\lambda_t) = -\varepsilon U\nabla l(x_j;\lambda_t). \tag{3.8}$$

And then, the adjustment rule using the loss function for the GMM parameter $\{w_i,\vec{\mu}_i,V_i\}$ are given as

$$w_i(t+1) = w_i(t) - \varepsilon(t)\nu_k\varphi_j\zeta_{j,i}, \tag{3.9}$$

$$\mu_{i,m}(t+1) = \mu_{i,m}(t) - \varepsilon(t)\nu_k\varphi_j\eta_{j,i,m}, \tag{3.10}$$

$$V_i(t+1) = (V_i(t)^{-1} - \varepsilon(t)\nu_k\varphi_j\rho_{j,i})^{-1}, \tag{3.11}$$

and

$$\nu_k = \alpha l_k(d_k)(1 - l_k(d_k)), \tag{3.12}$$

$$\zeta_{j,i} = f_i(x_j)/w_i - \frac{\sum_{i=1}^{M}(f_i(x_j)/w_i)}{M}, \text{ where } f_i(x_j) = w_i b_i(x_j), \tag{3.13}$$

$$\eta_{j,i,m} = -f_i(x_j)V_i^{-1}\left(x_{j,m} - \mu_{j,i,m}\right), \tag{3.14}$$

$$\rho_{j,i} = \frac{1}{2} f_i(x_j)[D \cdot V_i - \sum_{m=1}^{D}\left(x_{j,m} - \mu_{j,i,m}\right)^2], \tag{3.15}$$

for $j = k$,

$$\varphi_j = -1, \tag{3.16}$$

for $j \neq k$,

$$\varphi_j = \frac{1}{N-1}\left[\frac{1}{N-1}\sum_{n\neq k}^{N}(\frac{g_n}{g_j})^{\mu}\right]^{1/\mu - 1}. \tag{3.17}$$

The block of GPD-based GMM model for the training phase is shown in Fig. 3-4.

In the test phase, we use the misclassification measure to decide if the speaker is an imposter. When $d_k(x_j;\lambda_k)$ is larger, it represents the degree of misclassification is higher. On the other hand, when $d_k(x_j;\lambda_k)$ is smaller, it classifies the speaker more correctly. Therefore, the choice of the threshold is important. If the threshold is large, the rejection rate for some imposter will become low; if the threshold is small, the identification rate of a customer will be reduced. We must find the balance between the rejection rate and the identification rate.

The block diagram of GPD-based GMM model for the test phase is illustrated in Fig. 3-5.

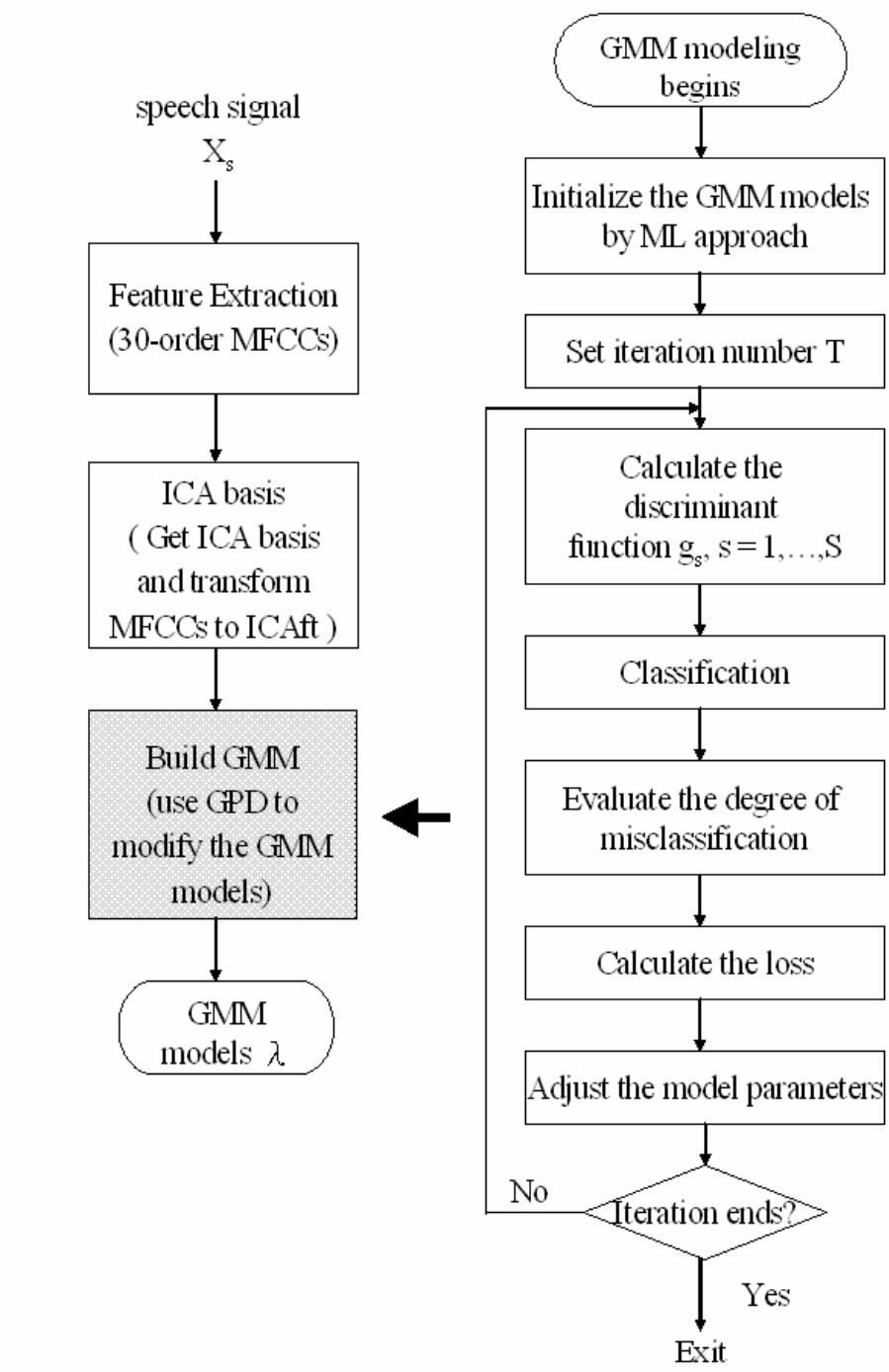



Fig. 3-4 Block diagram of GPD-based GMM model for the training phase.

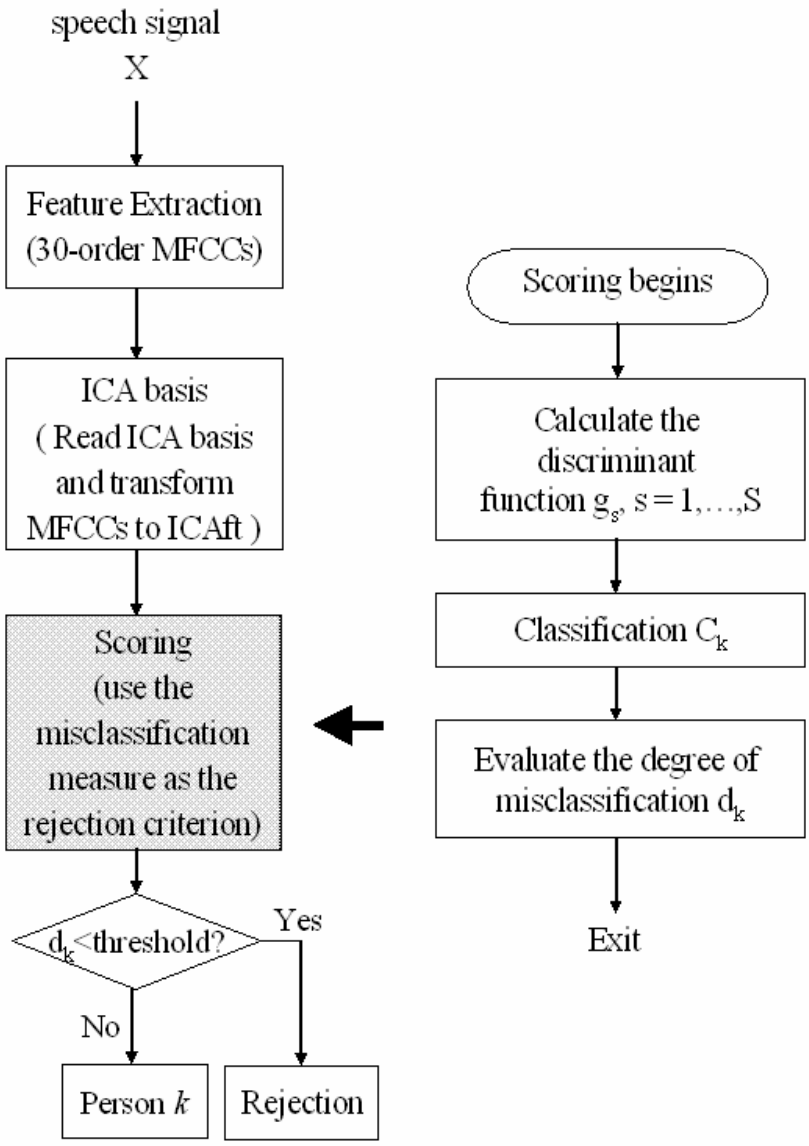



Fig. 3-5 Block diagram of GPD-based GMM model for the test phase.

# Chapter 4

# Experiment Results and Discussion

## 4.1 Introduction

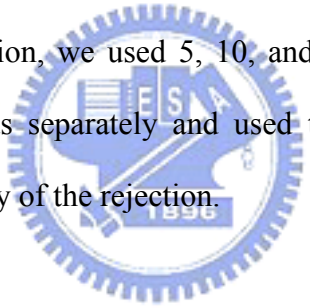
In the pervious chapter, we described the structures of the proposed speaker recognition system. For investigating and showing the contribution and efficiency of these methods we applied, several sets of experiments were done. In the first set of experiments, we evaluated the ML-based GMM with MFCC. The ML-based GMM with ICA features was evaluated in the second set of experiments. In these experiments, we tried to show the superior ability of the ICA features for the speaker recognition task. In the third set of experiments, we adopted the MFCC as the features and the proposed GPD-based GMM as the classifier. The improvement caused by the classifier optimization is shown here. Finally, in the forth set of experiments, we combined the ICA features and the GPD-based GMM as the overall speaker recognition system. The experimental results showed the contribution of this model.

For these experiments, several processing steps occur in the front-end speech analysis. First, the speech signal was decomposed in frames of 256 samples with an overlap of 128 samples (the sampling rate is 8k Hz). For each frame, FFT was computed and provided 256 square module values representing the short term power spectrum in the 0-4k Hz band. And then, this Fourier power spectrum was used to compute 16 mel-spaced filter bank coefficients. We finally computed the power accumulated in each filter bank and the discrete cosine transformation (DCT) to get

the cepstral coefficients called MFCCs with 30 orders.

## 4.2 Experiment Database

The database for the experiments is the TIMIT acoustic-phonetic speech corpus. This corpus is widely used throughout the world and provides a standard that permits direct comparison of experimental results obtained by different methodologies. In this thesis, we only used a subset of the DR2 from TIMIT database. This set represents 76 speakers of the same (North America) dialect. There are 52 males and 23 females in this set. The corpus consists of 10 sentences recorded from each speaker. We randomly choose 8 sentences to train the speaker models, and the other 2 sentences to test. For the speaker recognition, we used 5, 10, and 20 speakers as the customers from the DR2 speaker corpus separately and used the reminding speakers as the imposters to evaluate the utility of the rejection.



## 4.3 Experiment Result

In the following, four sets of experiments would be carried out to evaluate our recognition system.

We assigned one class to each set of features, and after the process of voting by the classifications of features, we could make sure which person the speaker was. The recognition rate was calculated by the result of the correct classification.

## Experiment I

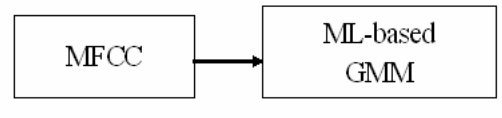



Fig. 4-1 Sketch of the feature and the GMM model of Experiment I

Table 4-1 Recognition Results of Experiment I for 5 customers (71 imposters)

| rejection threshold(*$10^3$) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 5 | 69 | 71 | 71 | 72 | 73 | 74 | 74 | 74 | 75 | 75 | 75 |
| Error True No. | 71 | 7 | 5 | 5 | 4 | 3 | 2 | 2 | 2 | 1 | 1 | 1 |
| Error False No. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 4-2 Recognition Results of Experiment I for 10 customers (66 imposters)

| rejection threshold(*$10^3$) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 10 | 66 | 68 | 70 | 69 | 69 | 71 | 72 | 74 | 74 | 74 | 74 |
| Error True No. | 66 | 10 | 8 | 6 | 6 | 6 | 4 | 3 | 0 | 0 | 0 | 0 |
| Error False No. | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |

Table 4-3 Recognition Results of Experiment I for 20 customers (56 imposters)

| rejection threshold(*$10^3$) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 20 | 66 | 67 | 68 | 68 | 70 | 70 | 72 | 73 | 71 | 69 | 67 |
| Error True No. | 56 | 9 | 8 | 7 | 7 | 5 | 5 | 3 | 2 | 3 | 4 | 5 |
| Error False No. | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 4 |

In these tables, “Right No.” means that the number of right classifications from 76 speaker; “Error True No.” represents that the number of false classifications from the imposters; “Error False No.” is the number of false rejections from the customers.

## Experiment II

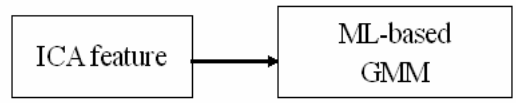



Fig. 4-2 Sketch of the feature and the GMM model of experiment II

Table 4-4 Recognition Results of Experiment II for 5 customers (71 imposters)

| rejection threshold(*$10^3$) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 5 | 70 | 71 | 72 | 72 | 74 | 74 | 74 | 74 | 75 | 76 | 76 |
| Error True No. | 71 | 6 | 5 | 4 | 4 | 2 | 2 | 2 | 2 | 1 | 0 | 0 |
| Error False No. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 4-5 Recognition Results of Experiment II for 10 customers (66 imposters)

| rejection threshold(*$10^3$) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 10 | 66 | 68 | 70 | 71 | 71 | 72 | 72 | 73 | 73 | 74 | 74 |
| Error True No. | 66 | 10 | 8 | 6 | 5 | 4 | 3 | 3 | 2 | 2 | 1 | 1 |
| Error False No. | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 4-6 Recognition Results of Experiment II for 20 customers (56 imposters)

| rejection threshold(*$10^3$) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 20 | 66 | 67 | 68 | 70 | 71 | 71 | 73 | 73 | 73 | 73 | 73 |
| Error True No. | 56 | 9 | 8 | 7 | 5 | 4 | 4 | 2 | 2 | 2 | 2 | 2 |
| Error False No. | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

## Experiment III

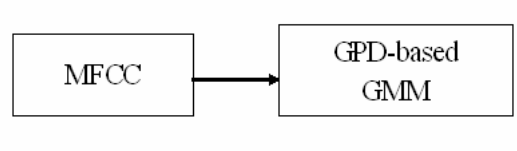



Fig. 4-3 Sketch of the feature and the GMM model of experiment III

Table 4-7 Recognition Results of Experiment III for 5 customers (71 imposters)

| rejection threshold(*10³) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 5 | 72 | 73 | 74 | 74 | 75 | 75 | 75 | 76 | 76 | 76 | 76 |
| Error True No. | 71 | 4 | 3 | 2 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Error False No. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 4-8 Recognition Results of Experiment III for 10 customers (66 imposters)

| rejection threshold(*10³) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 10 | 69 | 70 | 70 | 73 | 73 | 75 | 75 | 75 | 75 | 75 | 74 |
| Error True No. | 66 | 7 | 5 | 5 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Error False No. | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |

Table 4-9 Recognition Results of Experiment III for 20 customers (56 imposters)

| rejection threshold(*10³) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 20 | 69 | 70 | 70 | 70 | 71 | 71 | 73 | 73 | 72 | 73 | 73 |
| Error True No. | 56 | 6 | 5 | 5 | 5 | 4 | 4 | 2 | 2 | 2 | 1 | 1 |
| Error False No. | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |

## Experiment IV

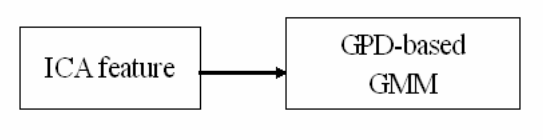



Fig. 4-4 Sketch of the feature and the GMM model of experiment IV

Table 4-10 Recognition Results of Experiment IV for 5 customers (71 imposters)

| rejection threshold(*10³) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 5 | 72 | 74 | 75 | 75 | 75 | 75 | 76 | 76 | 76 | 76 | 76 |
| Error True No. | 71 | 4 | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Error False No. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 4-11 Recognition Results of Experiment IV for 10 customers (66 imposters)

| rejection threshold(*10³) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 10 | 71 | 73 | 73 | 73 | 73 | 75 | 75 | 75 | 75 | 75 | 75 |
| Error True No. | 66 | 5 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Error False No. | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 4-12 Recognition Results of Experiment IV for 20 customers (56 imposters)

| rejection threshold(*10³) | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Right No. | 20 | 70 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 | 73 |
| Error True No. | 56 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Error False No. | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

We could see that if the rejection threshold is set to 0, then no one (includes customers and imposters) would be rejected; that is to say, the error false number is zero, the error false number equals to the imposter number, and the right classification number is equivalent to the customer number. Therefore, the recognition rate is worst when the rejection threshold is zero.

Besides, when the rejection threshold is larger, the more imposters were rejected. Hence, the recognition rate would also be raised. It means that the grades (probabilities) of the customers are greater than those of the imposters. But the customer might be rejected if the threshold was too large.

## Comparison

### 5 customers (71 imposters)

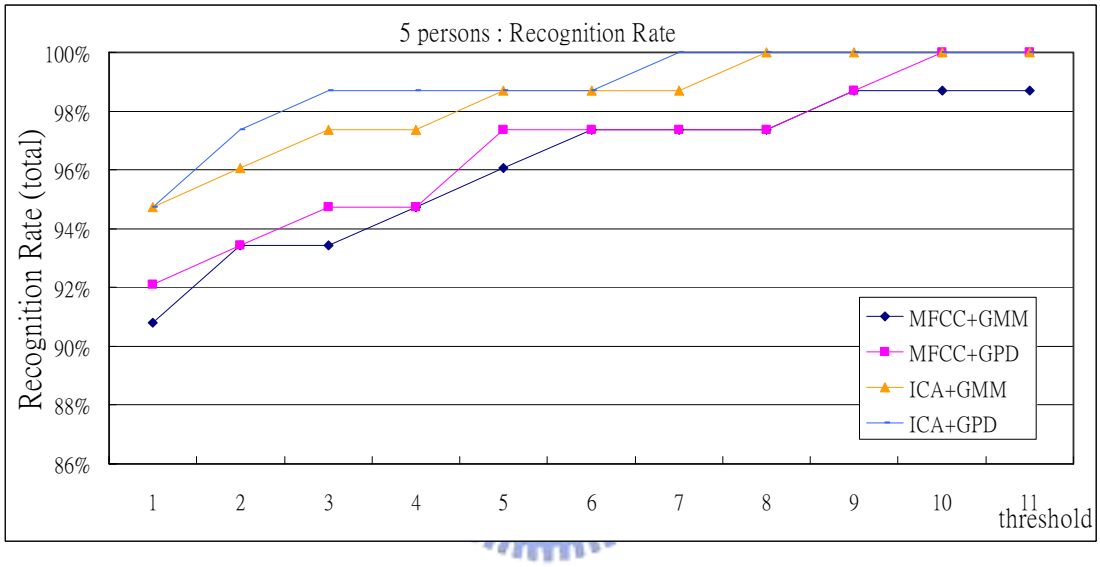



Fig. 4-5 the recognition rates of four experiments

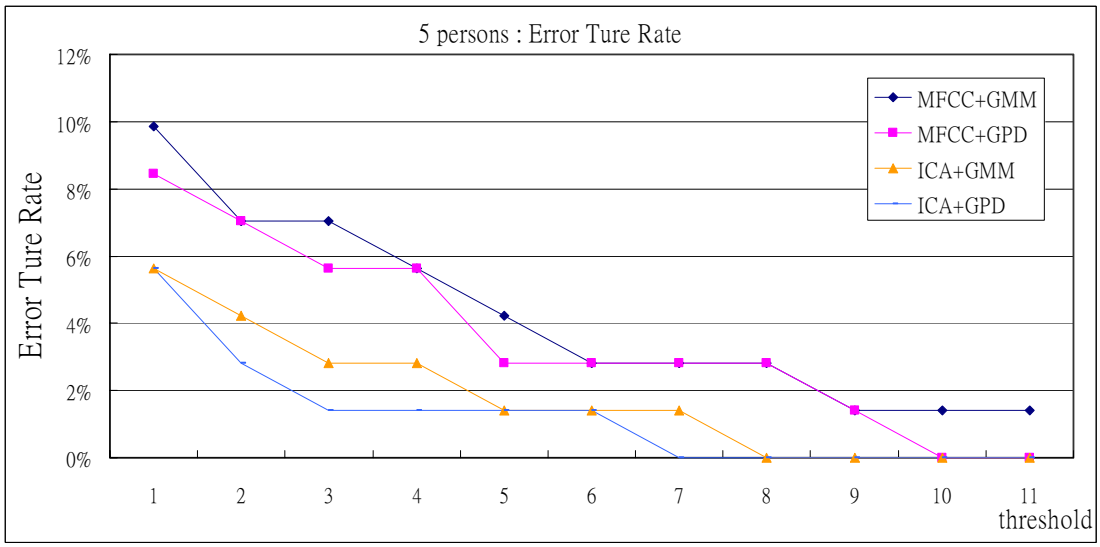



Fig. 4-6 the error true rates of four experiments


Fig. 4-7 the error false rates of four experiments

**10 customers (66 imposters)**

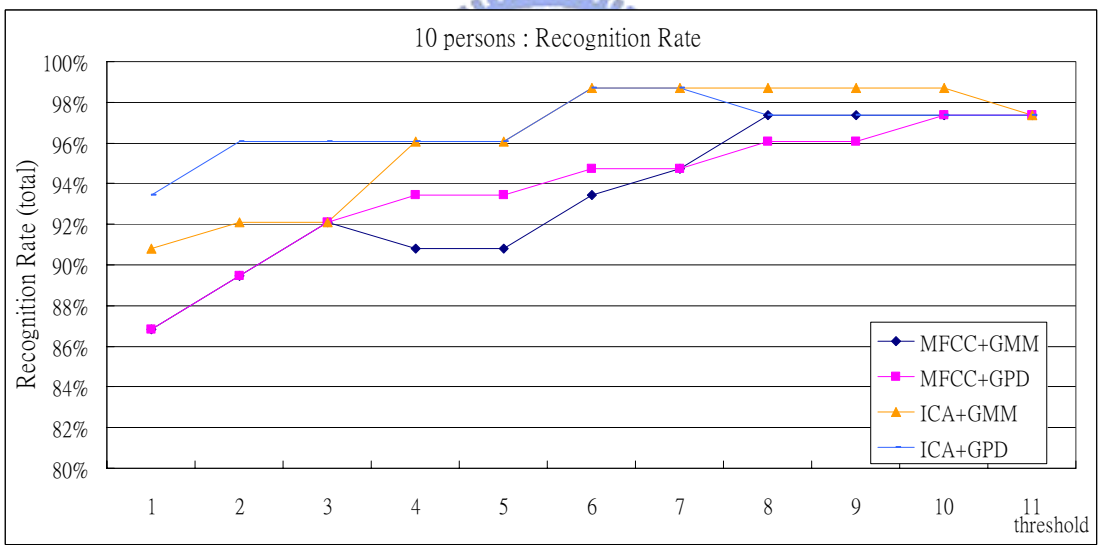



Fig. 4-8 the recognition rates of four experiments

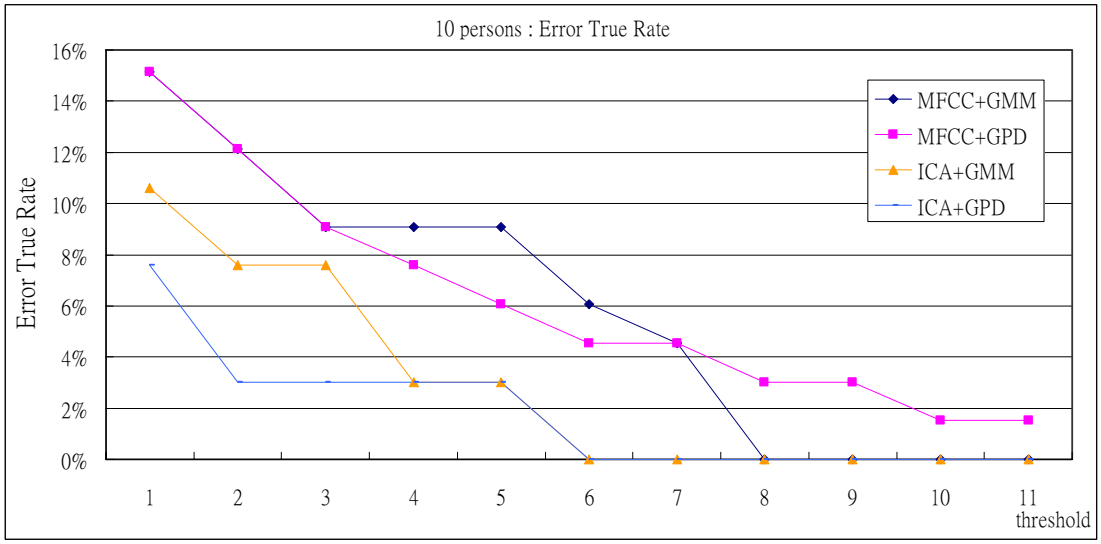



Fig. 4-9 the error true rates of four experiments


Fig. 4-10 the error false rates of four experiments

**20 customers (56 imposters)**


Fig. 4-11 the recognition rates of four experiments

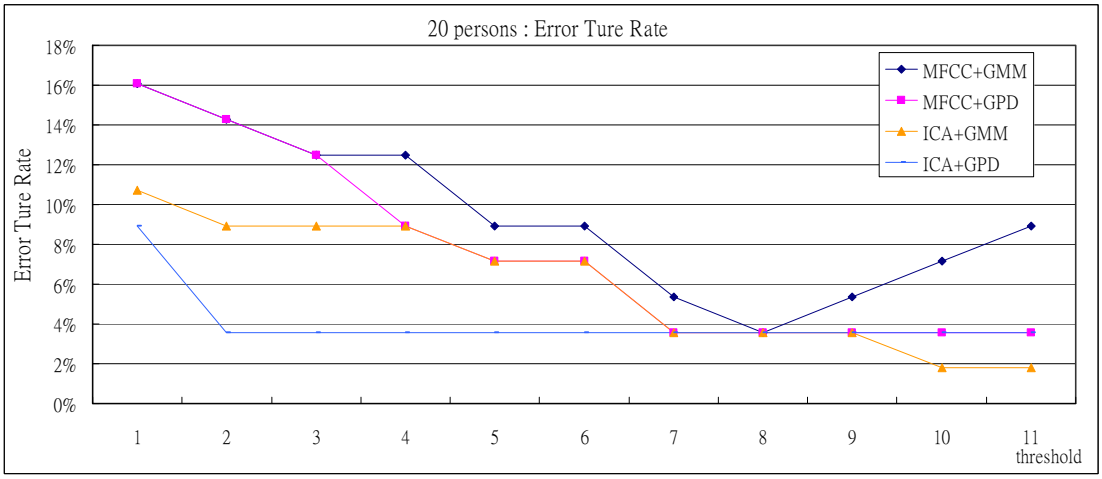



Fig. 4-12 the error true rates of four experiments

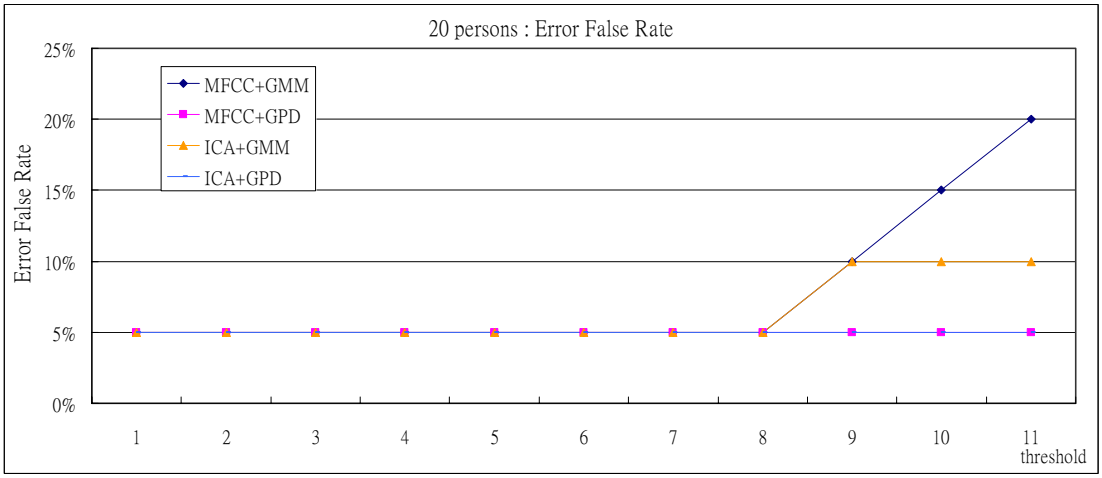



Fig. 4-13 the error false rates of four experiments

From the above figures, Experiment IV has the best performances in the recognition rate, the error true rate, and the error false rate; on the other hand, Experiment I has the worst recognition rate. In addition, the recognition rate is higher when the customers are fewer.

## 4.4 Discussion

By using ICA to transform MFCC to the independent basis, we could obtain a better feature for the GMM recognizer. And from the experiments, we observed that the performance of the GPD-based GMM was also better than that of the ML-based GMM. Therefore, we combined the two algorithms into our speaker recognition system, and then we could get the best recognition rate of all the four systems. It was proven that our proposed recognition system was really improved the conventional speaker recognition system.

# Chapter 5

# Conclusion and Future Work

## 5.1 Conclusion

In this thesis, we develop an text-independent speaker recognition system. It has two main subjects to construct the system. One is ICA used to find out the independent basis for transforming MFCC to the more important features and reducing the dimension. The other is the GPD optimizer applied to modify the GMM recognizer. We show the formulation of the GPD algorithm can be blended into the GMM recognizer design.

A series of experiments are conducted to examine the efficiencies of ICA and the GPD algorithm. Because the ICA-based features are contained the most important components in MFCC, it has better performance than that of MFCC. Besides, the new features transformed by the ICA basis has fewer dimensions, it can save computation. It showed in experiment I and experiment II.

A GPD algorithm is analyzed and applied to a conventional GMM-based speaker recognizer. We show that the formulation of the GPD algorithm is compatible with GMM, and we also present an implementation of the GPD method in a GMM-based speaker recognizer.

The experiments I~IV has shown the performance of the GPD-based GMM. Compared our proposed system (experiment IV) with the conventional system (experiment I), it is improved approximately 5%.

## 5.2 Future Work

By using ICA, we can find the hidden predictive information of the speech signals and reduce the dimension of the data. However, how many dimensions we select will have the best performance is the interesting problem. If we are able to know about it, we could raise the recognition rate and would not waste the operation. In other words, if we could know what each independent component represents, such as formants, pitches, and so on, then we can use them directly instead of choosing them empirically.

For GPD, a most important point is the discovery of a desirable form of the discriminant function. Solving this problem will advance the speaker recognition technology, but it is obviously difficult and needs significant research efforts. Another important point is to find a reasonable method of controlling the smoothness of the functions – the smooth classification error count loss for example.

In addition, GPD-based training suffers from a scaling problem; it means that extensive computation is involved in evaluating the interclass competition over the tremendous number of possible classes in a large-scale task, such as large-number speaker identification. This problem also occurs in the misclassification measure processing. It will cause the optimization used in GPD to be slower than the conventional method, ex. the expectation maximization (EM) method. Then the $L_\infty$ norm may be needed to reduce the adjustment computation in the training phase.

Besides, the success of the GPD method is depended on a good selection of some parameters which the designer decided, such as $\varepsilon$ and $\mu$. But the selection is usually performed experimentally due to a lack of theory, a more theoretically selection method is needed.

Finally, we can apply this speaker recognition system to the speech recognition

since they are kinds of recognition. Of course, it requires some modification between the two systems. For example, we should use HMM to replace GMM for continuous speech signals.

# Bibliography


[1] D. O'Shaughnessy, "Speaker Recognition," *IEEE ASSP Mag.*, pp. 4-17, Oct. 1986.

[2] Q. Li, B. H. Juang, C. H. Lee, Q. Zhou, and F. K. Soong, "Recent Advancements in Automatic Speaker Authentication", *IEEE Robotics & Automation Mag.*, pp. 24-34, Mar. 1999.

[3] D.A. Reynolds, R.C. Rose, and M. J. T. Smith, "PC-based TMS320C30 Implementation of the Gaussian Mixture Model Text-Independent Speaker Recognition System," in *Proc. Int. Conf. Signal Processing Applications Technol.*, pp. 967-973, Nov. 1992.

[4] J. Fortuna, D. Schuurman, and D. Capson, "A Comparison of PCA and ICA for Object Recognition under Varying Illumination," *Proc. of 2002 Int. Conf. on Pattern Recognition,* vol. 3, pp. 11-15, Aug. 2002.

[5] J. H. Lee, H. Y. Jung, T. W. Lee, and S. Y. Lee, "Speech Feature Extraction Using Independent Component Analysis," in *Proc. ICASSP*, Istanbul, Turkey, vol. 3, pp. 1631-1634, Jun. 2000.

[6] H. Y. Jung, M. Park, H. R. Kim, and M. Hahn, "Speaker Adaptation Using ICA-Based Feature Transformation," *ETRI Journal*, vol. 24, no. 6, pp. 469-472, Dec. 2002.

[7] V. Moonasar and G. K. Venayagamoorthy, "Speaker Identification Using a Combination of Different Parameters as Feature Inputs to an Artificial Neural Network Classifier," *Proc. of 1999 Int.Conf. on Africon,* vol. 1, pp. 189-194, 1999.

[8] T. Kinnunen and I. Karkkainen, "Class-Discriminative Weighted Distortion

Measure for VQ-Based Speaker Identification," *SSPR & SPR 2002, LNCS 2396*, pp. 681-688, 2002.

[9] R. Soganci, F. Gurgen, and H. Topcuoglu, "Parallel Implementation of a VQ-Based Text-Independent Speaker Identification," *ADVIS 2004, LNCS 3261*, pp. 291-300, 2004.

[10] C. Miyajima, Y. Hattori, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker Identification Using Gaussian Mixture Models Based on Multi-Space Probability Distribution," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, vol. 1, pp. 433-436, May 2001.

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

[12] O. W. Kwon and C. K. Un, "Discriminative Weighting of HMM State-Likelihoods Using the GPD Method," *IEEE Signal Processing Letters*, vol. 3, no. 9, pp. 257-259, Sep. 1996.

[13] M. Inman, D. Danforth, S. Hangai, and K. Sato,"Speaker Identification Using Hidden Markov Models," *Proc. of 1998 Int. Conf. on Signal Processing*, vol. 1, pp. 609-612, Oct. 1998.

[14] J. E. Higgins and R. I. Damper, "An HMM-Based Subband Processing Approach to Speaker Identification," *AVBPA 2001, LNCS 2091*, pp. 169-174, 2001.

[15] D. A. Reyonds, "Speaker Identification and Verification Using Gaussian Mixture Speaker Models," *Speech Communication*, vol. 17, pp. 91-108, 1995.

[16] D. A. Reynolds, "Large Population Speaker Identification Using Clean and Telephone Speech," *IEEE Signal Processing Letters*, vol. 2, no. 3, pp46-48, Mar. 1995.

[17] P. C. Chang and B. H. Juang, "Discriminative Training of Dynamic Programming Based Speech Recognizers," *IEEE Trans. on Speech and Audio Processing*, vol. 1, no. 2, pp.135-143, Apr. 1993.

[18] T. M. Cover and J. A. Thomas, "*Elements of Information Theory*," Wiley.

[19] A. Hyvärinen, "New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit, " in *Advances in Neural Information Processing System*s, vol. 10, pp. 273-279, 1998.

[20] S. Katagiri, C. H. Lee, and B. H. Juang, "A Generalized Probabilistic Descent Method," in *Proc. ASJ Autumn Conf.*, pp. 141-142, 1990.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Statist. Soc.*, vol. 39, no. 1, pp. 1-38, 1977.

[22] B. H. Juang and L. Rabiner, "The Segmental K-means Algorithm for Estimating Parameters of Hidden Markov Models," *IEEE Trans. Audio Speech Signal Processing*, vol. 38, pp. 1639-1641, Sep. 1990.

[23] S. Katagiri, B. H. Juang, and C. H. Lee, "Pattern Recognition Using a Family of Design Algorithms Based upon the Generalized Probabilistic Descent Method," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2345-2373, Nov. 1998.

[24] S. Katagiri, C. H. Lee, and B. H. Juang, "New Discriminative Training Algorithms Based on the Generalized Probabilistic Descent Method," in *Proc. IEEE Workshop Neural Networks for Signal Processing*, pp. 299-308, 1991.

[25] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Ann. Math. Stat.*, vol. 41, pp. 164-171, 1970.

[26] B. H. Juang and S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. Signal Processing*, vol. 40, pp. 3043-3054, Dec.

1992.

[27] W. Chou and B. H. Juang, "*Adaptive Discriminative Learning in Pattern Recognition*," Tech Rep., AT&T Bell Labs, Murray Hill, NJ.

[28] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 28, pp. 357-366, 1980.

[29] J. F. Cardoso and B. Laheld, "Equivariant Adaptive Source Separation," *IEEE Trans. Signal Processing*, vol. 45, no. 2, pp. 434-444, 1996.

[30] T. W. Lee, M. Girolami, A. J. Bell, and T. J. Sejnowski, "A Unifying Framework for Independent Component Analysis," *Computers and Math. with Applications*, vol.39,no. 11, pp.1-21, 2000.