Methods of Combining Multiple Classifiers with Different Features and Their Applications to Text-Independent Speaker Identification

Ke Chen, Lan Wang, and Huisheng Chi
National Lab of Machine Perception and Center for Information Science
Peking University, Beijing 100871, China
E-mail: {chen,lwang,chi}@cis.pku.edu.cn

International Journal of Pattern Recognition and Artificial Intelligence, 11(3), 1997, pp. 417-445.

Abstract

In practical applications of pattern recognition, there are often different features extracted from raw data which needs recognizing. Methods of combining multiple classifiers with different features are viewed as a general problem in various application areas of pattern recognition. In this paper, a systematic investigation has been made and possible solutions are classified into three frameworks, i.e. linear opinion pools, winner-take-all and evidential reasoning. For combining multiple classifiers with different features, a novel method is presented in the framework of linear opinion pools and a modified training algorithm for associative switch is also proposed in the framework of winner-take-all. In the framework of evidential reasoning, several typical methods are briefly reviewed for use. All aforementioned methods have already been applied to text-independent speaker identification. The simulations show that results yielded by the methods described in this paper are better than not only the individual classifiers' but also ones obtained by combining multiple classifiers with the same feature. It indicates that the use of combining multiple classifiers with different features is an effective way to attack the problem of text-independent speaker identification.

Keywords: Combination of multiple classifiers, different features, linear opinion pools, evidential reasoning, winner-take-all, maximum likelihood learning, EM algorithm, associative switch, text-independent speaker identification.

1 Introduction

Recently, the combination of multiple classifiers has been viewed as a new direction for the development of highly reliable pattern recognition systems, in particular, optical character recognition (OCR) systems. Preliminary results indicate that combination of several complementary classifiers leads to classifiers with improved performance [5, 53, 61, 68, 69]. There are at least two reasons justifying the necessity of combining multiple classifiers. First, for almost any one of the current pattern recognition application areas, there are a number of classification algorithms available developed from different theories and methodologies. Usually, for a specific application problem, each of these classifiers could reach a certain degree of success, but maybe none of them is totally perfect or at least anyone of them is not so good as expected in practical application. Second, for a specific recognition problem, there are often numerous types of features which could be used to represent and recognize patterns. These features are also represented in very diversified forms and it is rather hard to lump them together for one single classifier to make a decision. As a result, multiple classifiers are needed to deal with the different features [5, 68, 69, 71]. It also results in a general problem how to combine those classifiers with different features to yield the improved performance. The basic idea underlying the combination of multiple classifiers is that a consensus is made somehow based upon the results of multiple classifiers for a classification task using an elaborate combination scheme. So far, there have been extensive studies on the combination of multiple classifiers [1, 10, 20, 27, 35, 68]. Among these researches, possible solutions to the combination may be classified into three frameworks, i.e. linear opinion pools, winner-take-all and evidential reasoning. In the framework of *linear opinion pools*, the combination schemes make the final decision through the use of a linear combination of multiple classifiers' results. In regard to the linear coefficients or weights for combination, there are two kinds of methods for assigning values to linear coefficients or weights [35], i.e. weights as *veridical probabilities* [36, 45, 65, 67] and *minimum error weights* [16, 17, 30, 50]. In the framework of *winner-take-all*, a device called *associative switch* is used in the process of classification to choose the classification result of a specific classifier for a specific input pattern [69]. Since the combination scheme always chooses only one classifier among several classifiers to use its result as the final decision for a specific input pattern, the chosen classifier could be viewed as a winner and the style of the combination method is similar to the principle of winner-take-all in the unsupervised learning paradigm [31, 37]. In the framework of *evidential reasoning*, for an input pattern, the output of each individual classifier is regarded as an evidence or an event and the combination scheme makes the final decision based upon a method of evidential reasoning or a principle of voting [2, 5, 20, 26, 40, 53, 68].

The so-called *different features* problem refers to that there are numerous types of features which can be extracted from the same raw data for a classification task. Therefore, several different feature sets are available for a given data set. Based upon each one of these feature sets, a classifier or several different classifiers can be trained for the same classification task. It results in the existence of multiple classifiers with different features for the same classification task. As a result, the problem of combining multiple classifiers with different features is how to develop a scheme of combining these classifiers with different features to produce an improved result for the classification task. In the current techniques, it is found that most of methods in the framework of evidential reasoning could be directly applied to combine multiple classifiers with different features since the final decision is made merely by combining the classifiers' results using an evidential reasoning method or a voting principle regardless of the types of input (feature vector) to classifiers. In the framework of winner-take-all, the combination scheme, *associative switch*, could also combine multiple classifiers with different features by using the mapping or coding of different features as the input of the associative switch instead of different features themselves [69]. Using such techniques, the considerably better classification results have been produced in the field of OCR by combining multiple classifiers with different features [5, 68, 69]. In the framework of linear opinion pools, it is possible to directly use the methods with minimum error weights to combine multiple classifiers with different features since the weights could be achieved by performing regression merely based upon the information of classifiers' errors regardless of types of input (feature vector) to each classifier [16, 17, 30, 50]. Unfortunately, it is difficult to use the existing techniques with weights as veridical probabilities [36, 45, 65, 67] for handling the problem since in these methods the achievement of linear coefficients usually depends upon the input (feature vector) to each classifier. Recent researches show that the techniques with weights as veridical probabilities can achieve considerably good results in the applications to combination of multiple classifiers with the *same feature* [65, 67]. In this paper, we present a novel linear combination scheme with weights as veridical probabilities to extend the state-of-the-art techniques for combining multiple classifiers with *different features*. In addition, we also propose a modified training method for the associative switch [69] in the framework of winner-take-all in order to yield better performance.

Speaker identification is the process of determining from which of the registered speaker when a given utterance comes. Furthermore, speaker identification systems can be either *text-independent* or *text-dependent*. By text-independent, we refer to that the identification procedure should work for any text in either training or testing [28]. This is a different problem than text-dependent speaker identification, where the text in both training and testing is the same or is known. Speaker identification is a rather hard task since a speaker's voice always changes in time. In particular, text-independent speaker identification is more difficult than text-dependent speaker identification since a text-independent speaker identification system has to use elaborate techniques to capture the speaker's individual characteristics regardless of the contents carried in the speech, while in text-dependent speaker identification the use of simple template matching techniques can directly exploit the voice individuality associated with each phoneme or syllable. In this paper, only text-independent speaker identification is considered. There have been extensive studies in speaker identification so far. In general, the technique of speaker identification includes feature extraction and classification. With respect to feature extraction, many kinds of individual features covering from the characteristics of vocal cords to speech spectrum have already been investigated and turned out to be useful to speaker identification [3, 4, 18, 21, 22, 23, 24, 28, 34, 44, 60, 70]. Unfortunately, none of those features is perfect for robustness so that there is less agreement on which parameterization of the speech spectrum to use for features [18, 24, 28, 52, 49]. In addition, some researchers intended to lump two or more features together into a composite feature [24, 43, 44, 48]. However, the performance of the systems based upon the composite features was not significantly improved. Furthermore, to a certain extent, the use of composite features results in the *curse of dimensionality* problem. In particular, the problem becomes quite serious when the techniques of neural computing with time-delay [6, 8, 9, 13, 62] are used. On the other hand, several kinds of classifiers have been also applied in speaker identification [9, 18, 24, 28, 49, 63]. These classifiers include distance classifiers [3, 4, 25, 33, 42], neural network classifiers [6, 7, 8, 11, 12, 13, 14, 19, 32, 46, 47, 54] and classifiers based upon parametric or non-parametric density estimation [28, 29, 52, 57, 59]. Since there are many kinds of features and classifiers, speaker identification becomes a typical task which needs to combine multiple classifiers with different features for robustness. Unlike the aforementioned techniques used in speaker identification, both the proposed and some existing methods are systematically investigated in this paper for text-independent speaker identification by combining multiple classifiers with different features. Experimental results demonstrate the effectiveness of these combination methods and indicate that the use of combining multiple classifiers with different features is a promising way for a text-independent speaker identification system to yield the significantly improved performance.

The remainder of the paper is organized as follows. Section 2 presents a novel method of combining multiple classifiers with different features in the framework of linear opinion pools. Section 3 describes a modified training algorithm for the associative switch in the framework of winner-take-all and section 4 briefly reviews some existing combination methods in the framework of evidential reasoning for use in our work. Section 5 presents the applications of the aforementioned combination methods in text-independent speaker identification and illustrates experimental results. Conclusions are drawn in the final section.

2 A Linear Combination Method for Different Features

In this section, we present a novel method of combining multiple classifiers with *different features* on the basis of the work in [65, 67] which can merely combine multiple classifiers with the *same feature*. In the method, a generalized finite mixture model based upon different features is proposed and the corresponding learning algorithms are presented by maximum likelihood estimation with an EM algorithm.

2.1 A Generalized Finite Mixture Model for Different Features

For a sample D in a data set $\mathcal{S}_0$ with M classes, $\Gamma = \{1, 2, \cdots, M\}$, we assume that there are P $(P > 1)$ feature vectors which can be independently extracted from the sample D called $\mathbf{x}_1(D), \cdots, \mathbf{x}_P(D)$. For simplicity, hereafter, we shall rewrite these P feature vectors as $\mathbf{x}_1, \cdots, \mathbf{x}_P$. Accordingly, we may employ N $(N \geq P)$ classifiers, $e_1, \cdots, e_N$, to learn the classification task using features extracted from raw data in $\mathcal{S}_0$, respectively, in which the input of the classifier e_j is the feature vector $\mathbf{x}_{p_j}$ $(j = 1, \cdots, N;\ 1 \leq p_j \leq P)$. Given pattern classes with the labels C_i ($i = 1, \cdots, M$), we consider such classifiers that for an input $\mathbf{x}_{p_j}$, the output of a classifier e_j is as follows,

$$\vec{p}_j(\mathbf{x}_{p_j}) = [p_{j1}(\mathbf{x}_{p_j}), \cdots, p_{jM}(\mathbf{x}_{p_j})]^T, \;\; p_{jk}(\mathbf{x}_{p_j}) \geq 0, \;\; \sum_{k=1}^{M} p_{jk}(\mathbf{x}_{p_j}) = 1 \tag{1}$$

where $p_{jk}(\mathbf{x}_{p_j})$ denotes the probability that the sample D belongs to C_k recognized by the classifier e_j with the input vector $\mathbf{x}_{p_j}$, and $\mathbf{x}_{p_j}$ denotes a feature vector of the sample D with its form of representation being a vector, a string or whatever else. The direct instances of such classifiers include those based upon parametric or non-parametric density estimation. Indeed, there are some classifiers, e.g. distance classifiers or neural network classifiers, which output a vector $\vec{u}_j(\mathbf{x}_{p_j}) = [u_{j1}(\mathbf{x}_{p_j}), \cdots, u_{jM}(\mathbf{x}_{p_j})]^T$ without satisfying $u_{jk}(\mathbf{x}_{p_j}) \geq 0$ and $\sum_{k=1}^{M} u_{jk}(\mathbf{x}_{p_j}) = 1$. Fortunately, these outputs can be transformed into the form in Eq.(1) using a function $g(s)$ [65], that is,

$$p_{jk}(\mathbf{x}_{p_j}) = \frac{g[u_{jk}(\mathbf{x}_{p_j})]}{\sum_{t=1}^{M} g[u_{jt}(\mathbf{x}_{p_j})]} \quad k = 1, \cdots, M \tag{2}$$

where $g(s) \geq 0$. There are various forms of the function $g(s)$ such as $g(s) = s$, $g(s) = 1/s$, $g(s) = e^{-s}$ or $g(s) = s^2$, $g(s) = 1/s^2$, $g(s) = e^{-s^2}$ according to whether or not $u_{jk}(\mathbf{x}_{p_j}) \geq 0$ $(k = 1, \cdots, M)$.

For an input-output pair $\{\mathbf{x}_{p_j}, \mathbf{y}\}$, where $\mathbf{y} = [y_1, \cdots, y_M]^T$ and y_k being a binary value ($y_k \in \{0, 1\}$) as well as satisfying $\sum_{k=1}^{M} y_k = 1$, a classifier e_j with the input vector $\mathbf{x}_{p_j}$ specifies a distribution given by

$$P(\mathbf{y}|\mathbf{x}_{p_j}, \theta_j) = \prod_{k=1}^{M} [p_{jk}(\mathbf{x}_{p_j}|\theta_j)]^{y_k} \tag{3}$$

where θ_j is the set of parameters of the classifier e_j and has been already fixed after the classifier was trained on the data set $\mathcal{S}_0$. For a fixed $\mathbf{x}_{p_j}$, it is reduced to a generalized Bernoulli distribution, while for a fixed $\mathbf{y}$, we can achieve a distribution specified by one of $p_{jk}(\mathbf{x}_{p_j}|\vec{\theta}_j)$'s.

For combining multiple classifiers with different features, another data set $\mathcal{S}_1$ is necessary for training a combination scheme. For an input-output pair $(D, \mathbf{y})$ in $\mathcal{S}_1$, the feature vector $\mathbf{x}_{p_j}$ ($1 \leq j \leq P$) is used as the input of the classifier e_j. Moreover, we assume that there are priors $\Phi = \{\alpha_{ij}(\mathbf{x}_i), \beta_i(D)\}$ ($i = 1, \cdots, P;\ j = 1, \cdots, N$) for each classifier e_j $(j = 1, \cdots, N)$ so that we can define a generalized finite mixture distribution for the pair $(D, \mathbf{y})$ as

$$P(\mathbf{y}|D, \Phi) = \sum_{j=1}^{N} \Big[\sum_{i=1}^{P} \beta_i(D)\alpha_{ij}(\mathbf{x}_i)\Big] P(\mathbf{y}|\mathbf{x}_{p_j}, \theta_j) = \sum_{j=1}^{N}\sum_{i=1}^{P} \beta_i(D)\alpha_{ij}(\mathbf{x}_i) \prod_{k=1}^{M} [p_{jk}(\mathbf{x}_{p_j}|\theta_j)]^{y_k} \tag{4}$$

where $\alpha_{ij}(\mathbf{x}_i) \geq 0$, $\beta_i(D) \geq 0$, $\sum_{j=1}^{N} \alpha_{ij}(\mathbf{x}_i) = 1$, $\sum_{i=1}^{P} \beta_i(D) = 1$. $\mathbf{x}_{p_j}$ $(1 \leq p_j \leq P)$ still denotes the input vector of the classifier e_j. The generalized finite mixture distribution leads to a new combination scheme for dealing with different features. The combination scheme consists of P combination subschemes (respectively corresponding to P different features) in which $\alpha_{ij}(\mathbf{x}_i)$ refers to the linear coefficient produced by subscheme i based upon the the feature vector $\mathbf{x}_i$ for the classifier e_j, while $\beta_i(D)$ refers to the a priori probability that for the sample D the ith subscheme is used to produce the linear coefficients for making the final decision in the combination scheme.

In Eq.(4), those priors $\alpha_{ij}(\mathbf{x}_i) \in \Phi$ $(i = 1, \cdots, P; j = 1, \cdots, N)$ are conditional on input vectors $\mathbf{x}_i$ $(i = 1, \cdots, P)$. As a result, we assume that

$$\alpha_{ij}(\mathbf{x}_i) = \frac{\lambda_{ij} P(\mathbf{x}_i, \varphi_{ij})}{\sum_{r=1}^{N} \lambda_{ir} P(\mathbf{x}_i, \varphi_{ir})};\ \lambda_{ij} \geq 0,\ \sum_{j=1}^{N} \lambda_{ij} = 1,\ i = 1, \cdots, P,\ j = 1, \cdots, N. \tag{5}$$

where $P(\mathbf{x}_i, \varphi_{ij}) \geq 0$ is a parametric function and, in particular, can be given by Gaussian distribution

$$P(\mathbf{x}_i, \varphi_{ij}) = P(\mathbf{x}_i, \mathbf{m}_{ij}, \Sigma_{ij}) = \frac{1}{(2\pi)^{\frac{n_i}{2}} |\Sigma_{ij}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{m}_{ij})^T \Sigma_{ij}^{-1} (\mathbf{x}_i - \mathbf{m}_{ij})\} \tag{6}$$

where n_i is the dimension of $\mathbf{x}_i$ and $\varphi_{ij} = (\mathbf{m}_{ij}, \Sigma_{ij})$ is the set of all parameters in Gaussian distribution. In the combination scheme, thus, the information from the outputs of classifiers, the desire label $\mathbf{y}$ and different input vectors $\mathbf{x}_i$ $(i = 1, \cdots, P)$ are jointly considered for combination. In Eq.(4), however, $\alpha_{ij}(\mathbf{x}_i)$ and $\beta_i(D)$ $(i = 1, \cdots, P; j = 1, \cdots, N)$ are still unknown and need learning from samples in $\mathcal{S}_1$. We shall propose a maximum likelihood learning method to determine these priors in the sequel. Suppose that those priors have been already determined, for an unknown input sample D, $P(\mathbf{y}|D)$ can be computed by

$$P_k(D) = P(y_k = 1|D, \Phi) = \sum_{j=1}^{N} \sum_{i=1}^{P} \beta_i(D)\alpha_{ij}(\mathbf{x}_i)p_{jk}(\mathbf{x}_{p_j}); \quad k = 1, \cdots, M. \tag{7}$$

Using Eq.(7), the decision rule is defined as

$$E(D) = \begin{cases} k & \text{if } P_k(D) = \max_{1 \le i \le M} P_i(D) \ge T \\ M+1 & \text{otherwise} \end{cases} \tag{8}$$

where $0 < T < 1$ is a predefined threshold. $E(D)$ is the final decision and $E(D) = M + 1$ denotes that the sample D is rejected.

2.2 Maximum Likelihood Learning with EM Algorithm

For a classification task, we assume that there are N classifiers which have been trained with P different features extracted from raw data in the data set $\mathcal{S}_0$. Given another data set $\mathcal{S}_1 = \{(D^{(t)}, \mathbf{y}^{(t)})\ t = 1, \cdots, T\}$ (*observed data*), using P different features extracted from samples in $\mathcal{S}_1$, parameters $\Phi = \{\alpha_{ij}(\mathbf{x}_i), \beta_i(D)\}$ $(i = 1, \cdots, P; j = 1, \cdots, N)$ are estimated by maximizing the log-likelihood

$$L = \sum_{t=1}^{T} \log P(\mathbf{y}^{(t)}|D^{(t)}, \Phi) = \sum_{t=1}^{T} \log \Big[\sum_{j=1}^{N} \sum_{i=1}^{P} \beta_i(D^{(t)})\alpha_{ij}(\mathbf{x}_i^{(t)})P(\mathbf{y}^{(t)}|\mathbf{x}_{p_j}^{(t)}, \theta_j) \Big] \tag{9}$$

where $\mathbf{x}_{p_j}^{(t)}$ $(1 \le p_j \le P)$ denotes the input vector of classifier e_j at time t. For the log-likelihood, we adopt an EM algorithm [15] to estimate all parameters in Φ by introducing a set of indicators as *missing data* to observed data. For $i = 1, \cdots, P$ and $j = 1, \cdots, N$, these indicators are defined as

$$I_i^{(t)} = \begin{cases} 1 & \text{if the linear coefficients are determined by subscheme } i \\ 0 & \text{otherwise} \end{cases}$$

$$I_j^{(t)} = \begin{cases} 1 & \text{if } \mathbf{y}^{(t)} \text{ is generated from classifiers } e_j. \\ 0 & \text{otherwise} \end{cases}$$

where $\sum_{i=1}^{P} I_i^{(t)} = 1$ and $\sum_{j=1}^{N} I_j^{(t)} = 1$ ($t = 1, \cdots, T$). Thus, the complete data consists of both the observed data and the missing data.

An EM algorithm first finds the expected value of the complete-data likelihood. For the observed data and the proposed mixture model, the *Expectation step* (E-step) computes the following expectation of the complete log-likelihood at the sth iteration using Bayes' rule

$$\begin{aligned} E[I_i^{(t)}, I_j^{(t)}|\mathcal{X}] &= P(I_i^{(t)} = 1, I_j^{(t)} = 1|\mathbf{y}^{(t)}, D^{(t)}, \Phi^{(s)}) \\ &= \frac{P_j(\mathbf{y}^{(t)}|\mathbf{x}_{p_j}^{(t)}, \theta_j)\alpha_{ij}^{(s)}(\mathbf{x}_i^{(t)})\beta_i^{(s)}(D^{(t)})}{\sum_{j=1}^{N} \sum_{i=1}^{P} \beta_i^{(s)}(D^{(t)})\alpha_{ij}^{(s)}(\mathbf{x}_i^{(t)})P_j(\mathbf{y}^{(t)}|\mathbf{x}_{p_j}^{(t)}, \theta_j)} \end{aligned} \tag{10}$$

$$\begin{aligned} E[I_i^{(t)}|\mathcal{X}] &= P(I_i^{(t)} = 1|\mathbf{y}^{(t)}, D^{(t)}, \Phi^{(s)}) \\ &= \frac{\sum_{j=1}^{N} P_j(\mathbf{y}^{(t)}|\mathbf{x}_{p_j}^{(t)}, \theta_j)\alpha_{ij}^{(s)}(\mathbf{x}_i^{(t)})\beta_i^{(s)}(D^{(t)})}{\sum_{j=1}^{N} \sum_{i=1}^{P} \beta_i^{(s)}(D^{(t)})\alpha_{ij}^{(s)}(\mathbf{x}_i^{(t)})P_j(\mathbf{y}^{(t)}|\mathbf{x}_{p_j}^{(t)}, \theta_j)} \end{aligned} \tag{11}$$

That is, the a posteriori probabilities are obtained as follows,

$$h_i^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)}) = E[I_i^{(t)}|\mathcal{X}], \quad h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)}) = E[I_i^{(t)}, I_j^{(t)}|\mathcal{X}]. \tag{12}$$

To simplify the computation in the *Maximization step* (M-step), a trick in [66, 67] is used to rewrite Eq.(4) into the following equivalent form

$$P(\mathbf{y}, D) = P(\mathbf{y}|D, \Phi)P(\mathbf{x}_i, \varphi) = \sum_{j=1}^{N}\sum_{i=1}^{P} \beta_i(D)\lambda_{ij}P(\mathbf{x}_i, \varphi_{ij})P(\mathbf{y}|\mathbf{x}_{p_j}, \theta_j), \tag{13}$$

where $P(\mathbf{x}_i, \varphi) = \sum_{r=1}^{N} \lambda_{ir}P(\mathbf{x}_i, \varphi_{ir})$. Using Eq.(13), the task of Maximization step (M-step) is to solve the following separate optimal problems for $i = 1, \cdots, P,\ j = 1, \cdots, N$.

$$\varphi_{ij}^{(s+1)} = \arg\max_{\varphi_{ij}} \sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{i=1}^{P} h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)}) \log P(\mathbf{x}_i, \varphi_{ij}) \tag{14}$$

$$\lambda_{ij}^{(s+1)} = \arg\max_{\lambda_{ij}} \sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{i=1}^{P} h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)}) \log \lambda_{ij}; \quad s.t.\ \sum_{j=1}^{N} \lambda_{ij} = 1,\ \lambda_{ij} \geq 0. \tag{15}$$

$$\beta_i^{(s+1)} = \arg\max_{\beta_i} \sum_{t=1}^{T}\sum_{i=1}^{P} h_i^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)}) \log \beta_i; \quad s.t.\ \sum_{i=1}^{P} \beta_i = 1,\ \beta_i \geq 0. \tag{16}$$

Accordingly, the EM algorithm for training the proposed combination scheme is summarized as

1. **Initialization at s = 0**
   For $i = 1, \cdots, P$ and $j = 1, \cdots, N$, set $\beta_i^{(s)} = \frac{1}{P}$ and $\lambda_{ij}^{(s)} = \frac{1}{N}$ as well as initialize randomly $\varphi_{i1} = \varphi_{i2} = \cdots = \varphi_{iN}$ subject to $\alpha_{ij}^{(s)}(\mathbf{x}_i) = \frac{1}{N}$ $(j = 1, \cdots, N)$.

2. **The EM procedure at s $>$ 0**
   (1) **E-step**. For each pair $(D^{(t)}, \mathbf{y}^{(t)})$, compute the a posteriori probabilities $h_i^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)})$ and $h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)})$ $(i = 1, \cdots, P;\ j = 1, \cdots, N)$ using Eq.(10) and Eq.(11).
   (2) **M-step**. Find a new estimate for $i = 1, \cdots, P,\ j = 1, \cdots, N$ based upon the following updating formulae:

$$\beta_i^{(s+1)} = \frac{1}{T}\sum_{t=1}^{T} h_i^{(s)}(\mathbf{y}^{(t)}|D^{(t)}) \tag{17}$$

$$\mathbf{m}_{ij}^{(s+1)} = \frac{1}{\sum_{t=1}^{T} h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)})} \sum_{t=1}^{T} h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)})\mathbf{x}_i^{(t)} \tag{18}$$

$$\Sigma_{ij}^{(s+1)} = \frac{1}{\sum_{t=1}^{T} h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)})} \sum_{t=1}^{T} h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)})[\mathbf{x}_i^{(t)} - \mathbf{m}_{ij}^{(s+1)}][\mathbf{x}_i^{(t)} - \mathbf{m}_{ij}^{(s+1)}]^T \tag{19}$$

$$\lambda_{ij}^{(s+1)} = \frac{1}{T}\sum_{t=1}^{T} h_{ij}^{(s)}(\mathbf{y}^{(t)}|\mathbf{x}_i^{(t)}) \tag{20}$$

$$\alpha_{ij}^{(s+1)}(\mathbf{x}_i^{(t)}) = \frac{\lambda_{ij}^{(s+1)}P(\mathbf{x}_i, \mathbf{m}_{ij}^{(s+1)}, \Sigma_{ij}^{(s+1)})}{\sum_{r=1}^{N} \lambda_{ir}^{(s+1)}P(\mathbf{x}_i, \mathbf{m}_{ir}^{(s+1)}, \Sigma_{ir}^{(s+1)})} \tag{21}$$

3. Repeat Step 2 until a predefined ‘stop’ condition is satisfied.

3 A Modified Associative Switch for Different Features

For combining multiple classifiers, the basic idea underlying the combination in the framework of winner-take-all is to choose the one with the best result from N classifiers as the winner and to make the final decision by using its result for a specific sample if there is at least one classifier to classify the sample D correctly. When there is no classifier to classify the sample D correctly, the combination scheme will reject it. To complete the task of choosing a classifier for each sample, a device called *associative switch* has been proposed [69]. The associative switch is composed of N knobs sw_j $(j = 1, \cdots, N)$ with each sw_j installed on the output channel of classifier e_j to decide whether or not it is chosen as the winner for the current sample, i.e. whether or not to allow its output pass through to become the final decision. For an input sample D, thus, the output of each knob is given by

$$l_d = \begin{cases} l_j & \text{if } sw_j = \text{``}on'' \\ M+1 & \text{if } sw_j = \text{``}off'' \end{cases} \tag{22}$$

where l_d is the class label of the input sample D determined by the decision maker. Accordingly, N knobs are controlled by the output of the *winner-take-all combination mechanism* (WTA-CM). Assume that output of the WTA-CM is denoted as $\mathbf{c} = [c_1, \cdots, c_N]$ $(0 \leq c_j \leq 1;\ j = 1, \cdots, N)$. The behavior of each sw_j is determined as follows

$$sw_j = \begin{cases} \text{``}on'' & \text{when } j = \arg\max_{1 \leq k \leq N} c_k \text{ and } c_j \geq T \\ \text{``}off'' & \text{otherwise} \end{cases} \tag{23}$$

where T is a predefined threshold and usually $0.5 < T < 1$. To combine multiple classifiers with different features, for an unlabeled sample D, the WTA-CM also needs an input to determine the behavior of the switch. For this purpose, we adopt an encoding mechanism to produce a *mapping* or *coding* of the sample D , $M(D)$, as the input of the WTA-CM. Like the encoding mechanism in [69], here, $M(D)$ is just the label vector consisting of N labels produced by individual classifiers i.e. $M(D) = [l_1, \cdots, l_N]$ for the sample D. When different features of an unlabeled sample D are input to individual classifiers e_j $(j = 1, \cdots, N)$, $M(D)$ will also be input to the WTA-CM for recalling code **c** which determines the behavior of N knobs so as to either select one of outputs of N classifiers as the final label l_d or block all the output channel of individual classifiers and assign $M+1$ to l_d. Such a WTA-CM could be implemented by any existing artificial neural networks with the type of heteroassociative memory. In this paper, we simply use a three-layered multilayer perceptron (MLP) architecture to implement the WTA-CM.

The key task in the learning process of the associate switch is the design of the desired output for training the MLP used as the WTA-CM on a data set $\mathcal{S}_1$. Once we have the desired output the training procedure is the application of an existing learning algorithm such as *backpropagation algorithm* [55] on the training set $\mathcal{S}_1$. To design the appropriate desired output of the MLP used as the combination scheme, we first define a criterion for selecting a winner when there are several classifiers to give the correct result for a specific sample. We stipulate that the output of each classifier has the standard form in Eq.(1). If the output of a classifier does not satisfy the form in Eq.(1), it can be transformed into the standard form using Eq.(2). For a sample D, thus, each classifier e_j $(j = 1, \cdots, N)$ with the input vector $\mathbf{x}_{p_j}$ $(1 \leq p_j \leq P)$ has the output $\vec{p}_j(\mathbf{x}_{p_j}) = [p_{j1}(\mathbf{x}_{p_j}), \cdots, p_{jM}(\mathbf{x}_{p_j})]$. Among those M components of $\vec{p}_j(\mathbf{x}_{p_j})$, we can find two components in the following way

$$p_{jk_1}(\mathbf{x}_{p_j}) = \max_{1 \leq k \leq M} p_{jk}(\mathbf{x}_{p_j}), \quad p_{jk_2}(\mathbf{x}_{p_j}) = \max_{1 \leq k \leq M, k \neq k_1} p_{jk}(\mathbf{x}_{p_j}). \tag{24}$$

As a result, we define the criterion of selecting winner as follows,

$$j^* = \arg\max_{j \in \Lambda} \Delta p^{(j)}(\mathbf{x}_{p_j}), \quad \Delta p^{(j)}(\mathbf{x}_{p_j}) = p_{jk_1}(\mathbf{x}_{p_j}) - p_{jk_2}(\mathbf{x}_{p_j}). \tag{25}$$

where Λ is a set of all classifiers that output the correct label for a specific sample D. That is, the classifier e_{j^*} will be chosen as the winner. Assume that a sample D in $\mathcal{S}_1$ should belong to class $k \in \Gamma$ and classifier $e_j(\mathbf{x}_{p_j}) = l_j$ $(j = 1, \cdots, N;\ 1 \leq p_j \leq P)$ as well as $c_j^{(d)}[M(D)]$ is the jth component of the desire output of

WTA-CM for the sample D. Using the criterion defined in Eq.(25), we present a method to produce the desired output of the MLP used as the WTA-CM for the following three different cases:

Case 1. If $l_j \neq k$ for $j = 1, \cdots, N$, i.e. there is no individual classifier giving a correct classification, then we assign $c_j^{(d)}[M(D)] = 0$, $j = 1, \cdots, N$.

Case 2. If there is only one $l_{j^*} = k$ $(1 \leq j^* \leq N)$, i.e. there is only one individual classifier giving the correct classification, then for $j = 1, \cdots, N$, we let

$$c_j^{(d)}[M(D)] = \begin{cases} 1 & \text{when } j = j^* \\ 0 & \text{otherwise} \end{cases}$$

Case 3. If there is a subset $I_D \subseteq \{1, \cdots, N\}$ and $l_j = k$ for each $j \in I_D$, i.e. there is more than one individual classifier giving the correct result, then for $j = 1, \cdots, N$, we let

$$c_j^{(d)}[M(D)] = \begin{cases} 1 & \text{when } l_j = k \text{ and } j = \arg\max_{i \in I_D} \Delta p^{(i)}(\mathbf{x}_{p_i}) \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{x}_{p_i}$ $(1 \leq p_i \leq P)$ is a feature vector extracted from the sample D and used as the input vector of classifier e_i $(i = 1, \cdots, N)$. If there is more than one classifier in I_D satisfying both $j \in I_D$ and $j = \arg\max_{i \in I_D} \Delta p^{(i)}(\mathbf{x}_{p_i})$, moreover, then we arbitrarily or randomly choose only one of these classifiers as the winner, say j'. As a result, we let

$$c_j^{(d)}[M(D)] = \begin{cases} 1 & \text{when } l_{j'} = k \text{ and } j' = \arg\max_{i \in I_D} \Delta p^{(i)}(\mathbf{x}_{p_i}) \\ 0 & \text{otherwise} \end{cases}$$

4 Combination Methods Based on Evidential Reasoning

The combination methods based upon evidential reasoning have been extensively studied and already applied in the field of OCR [5, 53, 61, 68, 69]. The basic idea underlying the methods is that the result of each individual classifier is regarded as an evidence or an event and the final decision is made by consulting all combined classifiers with a method of evidential reasoning or evidence integrating. The methods of evidential reasoning (integrating) are usually based upon voting principle, Bayesian theory and Dempster-Shafer evidence theory. In this section, we briefly review some methods in [68] which have been applied to the experiments of text-independent speaker identification reported in this paper. In the sequel, the original representation in [68] will be rewritten for the purpose of combining multiple classifiers with different features.

4.1 A Combination Method in Bayesian Formalism

In order to combine multiple classifiers with different features in Bayesian formalism, the error of each classifier must be taken into consideration. As a result, the error of each classifiers e_j may be described by its *confusion matrix* [68], PT_j, as follows,

$$PT_j = \begin{bmatrix} n_{11}^{(j)} & n_{12}^{(j)} & \cdots & n_{1M}^{(j)} & n_{1(M+1)}^{(j)} \\ n_{21}^{(j)} & n_{22}^{(j)} & \cdots & n_{2M}^{(j)} & n_{2(M+1)}^{(j)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ n_{M1}^{(j)} & n_{M2}^{(j)} & \cdots & n_{MM}^{(j)} & n_{M(M+1)}^{(j)} \end{bmatrix} \tag{26}$$

for $j = 1, \cdots, N$; where each row i corresponds to class i, C_i, and each column l corresponds to the event $e_j(\mathbf{x}_{p_j}) = l$ $(l \in \Gamma \bigcup \{M+1\})$. Thus, an element $n_{il}^{(j)}$ denotes that $n_{il}^{(j)}$ samples belonging to C_i have been assigned a label l by classifier e_j. For an event $e_j(\mathbf{x}_{p_j}) = l$ of an error-bearing classifier e_j, its truth has uncertainty. With the knowledge of its confusion matrix PT_j, such an uncertainty could be modeled by the conditional probabilities that propositions $D \in C_i$ $(i = 1, \cdots, M)$ are true under the occurrence of the event $e_j(\mathbf{x}_{p_j}) = l$ is as follows,

$$P(D \in C_i | e_j(\mathbf{x}_{p_j}) = l) = \frac{n_{il}^{(j)}}{\sum_{i=1}^{M} n_{il}^{(j)}} \tag{27}$$

With N classifiers $e_1, \cdots, e_N$, we shall have N matrices $PT_1, \cdots, PT_N$. When these classifiers are used on feature vectors extracted from the sample D, N events $e_j(\mathbf{x}_{p_j}) = l_j$ $(j = 1, \cdots, N;\ l_j \in \Gamma \bigcup \{M+1\})$ will happen. An integrated belief value $bel(\cdot)$ is defined according to Bayesian formula and the conditional probabilities [68] as follows,

$$bel(i) = \frac{\prod_{j=1}^{N} P(D \in C_i | e_j(\mathbf{x}_{p_j}) = l_j)}{\sum_{i=1}^{M} \prod_{j=1}^{N} P(D \in C_i | e_j(\mathbf{x}_{p_j}) = l_j)} \tag{28}$$

where $\sum_{i=1}^{M} bel(i) = 1$ since $D \in C_i$ $(i = 1, \cdots, M)$ are mutually exclusive and exhaustive. $bel(i)$ becomes the combination scheme which collects evidence of combined classifiers with different features and integrates them for making final decision. Depending upon these $bel(i)$ values, therefore, the final decision is made by combining multiple classifiers with different features as follows,

$$E(D) = \begin{cases} k & \text{if } bel(k) = \max_{i \in \Gamma} bel(i) \geq T \\ M+1 & \text{otherwise} \end{cases} \tag{29}$$

where $0 < T \leq 1$ is a predefined threshold.

4.2 A Combination Method Based on Dempster-Shafer Theory

The Dempster-Shafer (D-S) theory of evidence [58] has been applied for combining multiple classifiers [41, 53, 68]. In the method used in our work, the combination is made in the situation that the recognition, substitution and rejection rates of each individual classifier are merely necessary as prior knowledge [68].

In the current problem, there are the M exhaustive and mutually exclusive propositions $A = D \in C_i$ $\forall i \in \Gamma$, which denote that an input sample D comes from class i with the label C_i, and the universal proposition is $\Theta = \{A_1, \cdots, A_M\}$. When applied to P $(1 \leq P \leq N)$ feature vectors extracted from the sample D, N classifiers, say $e_1, \cdots, e_N$, will produce N evidences $e_j(\mathbf{x}_{p_j}) = l_j$ $(j = 1, \cdots, N;\ 1 \leq p_j \leq P)$ with each classifier $e_j(\mathbf{x}_{p_j}) = l_j$ denoting that the sample D is assigned a label $l_j \in \Gamma \bigcup \{M+1\}$ by classifier e_j with the input vector $\mathbf{x}_{p_j}$. Given that γ_j, ϵ_j are the recognition rate and the substitution rate of classifier e_j, respectively. Usually $\gamma_j + \epsilon_j$ is less than one due to the rejection action. For each classifier $e_j = l_j$, if $l_j \in \Gamma$, one could have uncertain beliefs that the proposition $A_{l_j} = D \in C_{l_j}$ is true with a degree γ_j and is false with a degree ϵ_j; if $l_j = M+1$ (i.e. D is rejected by classifier e_j with the input vector $\mathbf{x}_{p_j}$), one has no knowledge about anyone of the M propositions $A_i = D \in C_i$, $\forall i \in \Gamma$, which could be viewed as the full support of the universal proposition Θ. To combine multiple classifiers with different features, we must discard some unnecessary evidences and special cases which include the evidences $e_j(\mathbf{x}_{p_j}) = l_j$ with $l_j = \{M+1\}$ and the cases of the recognition rate $\gamma_j = 1$ and the substitution rate $\epsilon_j = 1$. All these cases make the combination unnecessary. After ruling out the aforementioned evidences and special cases, we can concentrate on the general cases that there are N' evidences $e_j(\mathbf{x}_{p_j}) = l_j$ with $0 < \gamma_j < 1,\ 0 \leq \epsilon_j < 1$ $(j = 1, \cdots, N')$. For combination, we first collect the evidences into groups with those impacting the same proposition in each group, and then combine the evidences in the each group, respectively. Let us denote $\mathbf{m}(\cdot)$ and $bel(\cdot)$ as *basic probability assignment* (BPA) function and *belief* value in the D-S theory [58], respectively. For all the evidences $e_j(\mathbf{x}_{p_j}) = l_j$ $(j = 1, \cdots, N')$, suppose that among $j_1, \cdots, j_{N'}$ there are $N_1 \leq \min(M, N')$ different labels, say $l'_1, \cdots, l'_{N_1}$, thus all the N' evidences are divided into N_1 groups, say $E_1, \cdots, E_{N_1}$, in which each $e_j(\mathbf{x}_{p_j}) = l_j$ is put to group E_k if $e_j(\mathbf{x}_{p_j}) = l_j = l'_j$. For each group E_j, a combined BPA $\mathbf{m}_{E_j}$ can be obtained by recursively applying the combination rule in the D-S theory [58] to BPA's $\mathbf{m}_{j_1}, \cdots, \mathbf{m}_{j_r}$ provided by $e_{j_1}(\mathbf{x}_{p_{j_1}}), \cdots, e_{j_r}(\mathbf{x}_{p_{j_r}})$ since all evidences $e_{j_1}(\mathbf{x}_{p_{j_1}}) = l'_j, \cdots, e_{j_r}(\mathbf{x}_{p_{j_r}}) = l'_j$. That is,

$$\mathbf{m}_2 = \mathbf{m}_1 \oplus \mathbf{m}_{j_2},\ \mathbf{m}_3 = \mathbf{m}_2 \oplus \mathbf{m}_{j_3},\ \cdots,\ \mathbf{m}_r = \mathbf{m}_{r-1} \oplus \mathbf{m}_{j_r}. \tag{30}$$

Next, we further combine the BPA's $\mathbf{m}_{E_j}$ $(j = 1, \cdots, N_1)$ into a final combined BPA

$$\mathbf{m} = \mathbf{m}_{E_1} \oplus \mathbf{m}_{E_2} \oplus \cdots \oplus \mathbf{m}_{E_{N_1}} \tag{31}$$

and then to calculate the corresponding $bel(A_i)$ and $bel(\neg A_i)$ for $\forall i \in \Gamma$ based upon the final BPA **m**. The combination in Eq.(31) can be calculated with a fast computing method in [68].

On the basis of the belief values, $bel(A_i)$ and $bel(\neg A_i)$ $(i = 1, \cdots, M)$, the decision rule is defined as

$$E(D) = \begin{cases} k & \text{if } \Delta bel(A_k) = \max_{i\in\Gamma} \Delta bel(A_i) \geq T \\ M+1 & \text{otherwise} \end{cases} \tag{32}$$

where $0 < T < 1$ and $\Delta bel(A_i) = bel(A_i) - bel(\neg A_i)$ which reflects the pure total support by the proposition A_i.

4.3 A Combination Method Using Voting Principle

The committee voting principle is a general method to make a consensus by consulting several opinions. If each opinion could be viewed as an evidence, then the process of making a consensus will be regarded as the process of evidential reasoning. There have been several combination methods based upon the different voting principles [61, 68]. For a sample D, each classifier e_j $(j = 1, \cdots, N)$ produces a result of classification based upon one kind of feature extracted from the sample D, say $\mathbf{x}_{p_j}$ $(1 \leq p_j \leq P)$, i.e. $e_j(\mathbf{x}_{p_j}) = i$. We consider the event $e_j(\mathbf{x}_{p_j}) = i$ as an evidence and represent it with following form

$$E_j(D \in C_i) = \begin{cases} 1 & \text{if } e_j(\mathbf{x}_{p_j}) = i \text{ and } i \in \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

According to the *majority voting principle* that the consensus is made only if there are more than a half of people in the committee who have the same opinion, the decision rule is defined as

$$E(D) = \begin{cases} k & \text{if } E_S(D \in C_k) = \max_{i\in\Gamma} E_S(D \in C_i) > T \\ M+1 & \text{otherwise} \end{cases} \tag{34}$$

where $\frac{N}{2} \leq T < N$ is a predefined threshold and

$$E_S(D \in C_i) = \sum_{j=1}^{N} E_j(D \in C_i), \quad i = 1, \cdots, M. \tag{35}$$

5 Applications of Combination Methods to Text-Independent Speaker Identification

In this section, we present applications of all combination methods described in this paper to text-independent speaker identification. First, we describe the speech database and feature selection. Then individual classifiers used in the combination is described. It is followed by the results of individual classifiers and a specific combination. As a case study, finally, the results for comparison are also reported.

5.1 Speech Database, Feature Selection and Performance Evaluation

5.1.1 The Database

There is no standard database (benchmark) to evaluate speaker identification systems [44], though the DARPA TIMIT database which was originally designed to evaluate automatic speech recognition systems is often borrowed to evaluate speaker identification systems. The database for experiments reported in this paper is a subset of the standard speech database in China. This set represents 20 speakers of the same (Mandarin) dialect. Unlike the DARPA TIMIT database in which all utterances were recorded in the same session, the utterances in the database were recorded during three separate sessions. In the first session, 10 different phonetically rich sentences were uttered by each speaker. The average length of the sentences is about 4.5 seconds. In the second and the third sessions, 5 different sentences are uttered by each speaker, respectively. The average length of the sentences recorded in the second and the third sessions is about 4.4 and 5.0 seconds, respectively. All utterances were recorded in a quiet room and sampled at 11.025 kHz sampling frequency in 16 bit precision.

Some researchers have used the TIMIT database to evaluate their speaker identification systems and achieved the identifying accuracies close to 100% [6, 7, 8, 32]. However, it is not sufficient to claim that such systems are robust since there is little variation of speakers' characteristics carried in voices recorded in the same session. Actually, the performance of a speaker identification system should be evaluated by testing utterances recorded in different sessions [4, 22, 28, 29, 43, 44, 52, 59, 60]. As a result, in the experiments, the training set or Set-1 consists of 10 sentences recorded in the first session to train all individual classifiers. In addition, 5 sentences recorded in the second session are used as the training data or *cross-validation* data (Set-2) to train the combination schemes except the one based upon the voting principle and the test set or Set-3 is composed of 5 sentences recorded in the third session for testing both individual classifiers and combination methods described in the paper.

5.1.2 Feature Selection

Although *Wolf* outlined a set of desirable attributes on the chosen features for speaker recognition [64] more than 20 years ago, unfortunately, it is highly unlikely to find any set of features which simultaneously has all those attributes in practice [4, 18, 21, 22, 24, 28, 44, 49, 52, 56]. As a result, several features have already been investigated [4, 21, 22, 28, 34, 63]. The main outcome of the many feature selection studies was that features which represent *pitch* and the *speech spectrum* were the most effective for speaker identification. However, there is less agreement on which parameterization of the speech spectrum to use for features. Common spectrum representations for speaker identification are linear predictive coefficients and their various transformations (cepstral coefficients and PARCOR coefficients etc.) as well as the cepstrum and its variants such as Mel-scale cepstrum [4, 52, 56, 64]. As a result, we select four common features for the experiments, i.e. *linear predictive coding coefficients* (LPCC), *linear predictive coding cepstrum* (LPC-CEP), *cepstrum* (CEPS) and *Mel-scale cepstrum* (MEL-CEP) [51].

On the other hand, it is generally agreed that the voiced parts of an utterance, especially vowels and nasal, are more effective than the unvoiced parts for text-independent speaker identification [4, 52, 56, 64]. In experiments, therefore, only the voiced parts of a sentence are kept regardless of their contents by using a simple energy measuring method. The length of the Hamming analysis window is 64 ms without overlapping. It should be noted that the size of the analysis window is slightly larger than the commonly used sizes (normally $16 \sim 32$ ms) since it has been found that the identification performance is degraded with a normal analysis window [32]. Whenever the short-time energy of a frame of the sentence is higher than a predefined threshold, spectral features will be calculated. Furthermore, the samples are pre-emphasized by the filter $H(z) = 1 - 0.97z^{-1}$ and *24-order* LPCC, *24-order* LPC-CEP, *20-order* CEPS and *20-order* MEL-CEP are derived from the processed samples. For utterances of all 20 speakers, total numbers of feature vectors are 10057 frames in Set-1, 4270 frames in Set-2 and 4604 frames in Set-3, respectively.

5.1.3 Performance Evaluation

The evaluation of a speaker identification experiment is conducted in the following manner [52]. After feature extraction, the test speech is to produce a sequence of feature vectors denoted as $\{\vec{f}_1, \cdots, \vec{f}_t\}$. The sequence of feature vectors is divided into overlapping segments of S feature vectors. The first two segments from a sequence would be

$$\overbrace{\vec{f}_1, \vec{f}_2, \cdots, \vec{f}_S}^{\text{Segment 1}} \vec{f}_{S+1}, \vec{f}_{S+2}, \cdots \qquad \vec{f}_1, \overbrace{\vec{f}_2, \vec{f}_3, \cdots, \vec{f}_S, \vec{f}_{S+1}}^{\text{Segment 2}} \vec{f}_{S+2}, \cdots$$

A test segment length of 6.4 seconds would correspond to $S = 100$ feature vectors for a 6.4 ms frame rate. In the experiments reported in this paper, we choose $S = 100$; accordingly, total numbers of segments are 2290 in Set-2 and 2624 in Set-3, respectively, for utterances of all 20 speakers. Each segment of S vectors is treated as a separate test utterance and identified using the classification procedures of either individual classifiers or the combination of multiple classifiers. Using a segment, the system produces either an identifying result or a rejection. The above steps are repeated for test utterances from each speaker in the population. The final performance evaluation is then computed according to the identifying rate, rejection rate and substitution rate. In the sequel, *Identification, Substitution and Rejection* is the abbreviations for *identifying rate, substitution rate* and *rejection rate.* Accordingly, the *Reliability* is defined as

$$\text{Reliability} = \frac{\text{Identification}}{100\% - \text{Rejection}} \tag{36}$$

In the experiments, each speaker has approximately equal amount of testing speech so that the performance evaluation is not biased to any particular speaker.

5.2 Individual Classifiers

Given a sequence of feature vectors, $\{\vec{x}_s\}$, produced from an unknown speaker, the next task of the speaker identification system is to classify that sequence as having come from one of the speakers in the known population. As mentioned in the introduction of this paper, there are various classifiers which have already been used in speaker identification [6, 7, 11, 13, 19, 25, 28, 32, 33, 46, 47, 52, 54, 59]. For the same purpose as the selection of common features, we choose four benchmark classifiers commonly used in speaker identification [9, 18, 24, 28], i.e. *distance classifier, vector quantization, multilayer perceptron* and *Gaussian mixture model.*

5.2.1 The Distance Classifier

The *long term averaging* was an early method widely adopted for text-independent speaker identification. the basic idea underlying the methods is the comparison of an average computed on test data to a collection of stored averages developed for each of the speakers in training [42]. As a result, the distance classifiers play a prominent role for classification in the methods. In the methods, each speaker's voice characteristics are modeled by the average over all the feature vectors obtained from samples of the person's voice (training vectors), $\{\vec{x}_t^{(i)}\}_{t=1}^{T_i}$ $(i = 1, \cdots, K)$, as such, $\vec{\mu}^{(i)} = \frac{1}{T_i} \sum_{t=1}^{T_i} \vec{x}_t^{(i)}$. Then for classification, the average feature vector over the complete test utterance, $\{\vec{x}'_s\}_{s=1}^{S}$, is computed as $\vec{m} = \frac{1}{S} \sum_{s=1}^{S} \vec{x}'_s$ and compared to each speaker's model using a distance classifier as follows,

$$d(\vec{m}, \vec{\mu}^{(i)}) = (\vec{m} - \vec{\mu}^{(i)})^T W^{(i)} (\vec{m} - \vec{\mu}^{(i)}); \quad i = 1, \cdots, K. \tag{37}$$

where $W^{(i)}$ is a matrix used to allow different weighings to different directions in the feature space. For a reference group of K speakers, the test utterance is identified with speaker k only if $k = \arg\min_{1 \leq i \leq K} d(\vec{m}, \vec{\mu}^{(i)})$. With respect to the matrix $W^{(i)}$ in Eq.(37), there are various forms which result in the existence of multiple distance classifiers [3, 25, 28, 42]. In the experiments reported in this paper, the matrix $W^{(i)}$ has the following form:

$$W^{(i)} = \Sigma_i^{-1}, \quad \Sigma_i = \frac{1}{T_i}\sum_{t=1}^{T_i}(\vec{x}_t^{(i)} - \vec{\mu}^{(i)})(\vec{x}_t^{(i)} - \vec{\mu}^{(i)})^T; \quad i = 1, \cdots, K. \tag{38}$$

5.2.2 Vector Quantization

As a non-parametric model, the *vector quantization* (VQ) classifier was applied to speaker identification [59, 38] and has since been the benchmark classifier for text-independent speaker identification systems. Typically, a speaker is modeled by a VQ codebook of $32 \sim 128$ vectors derived using the LBG algorithm [39]. The clustering and recognition are carried out using the distance measure in Eq.(37) with the matrix $W^{(i)}$ $(i = 1, \cdots, K)$ in Eq.(38). The distance between a test vector $\vec{x}'_s$ and the ith speaker's codebook of C template vectors, $\{\vec{\mu}_1, \cdots, \vec{\mu}_C\}$, is the distance of $\vec{x}'_s$ to the closest template,

$$d_i(\vec{x}'_s) = \min_{1 \leq j \leq C} d(\vec{x}'_s, \vec{\mu}_j) \tag{39}$$

The implicit segmentation and acoustic class comparison are performed simultaneously by finding a closest template in a speaker's codebook and using that distance as the speaker similarity measure. Classification of a sequence of test vectors, $\{\vec{x}'_s\}_{s=1}^S$, is done by finding the speaker codebook producing the minimum average distance, which for the ith speaker's codebook is defined as

$$\bar{d}_i = \frac{1}{S}\sum_{s=1}^{S} d_i(\vec{x}'_s); \quad i = 1, \cdots, K. \tag{40}$$

For a reference group of K speakers, the test utterance is identified with speaker k only if $k = \arg\min_{1 \leq i \leq K} \bar{d}_i$. In the experiments, for each speaker, the codebook consists of 32 vectors and the matrix $W^{(i)}$ $(i = 1, \cdots, K)$ in Eq.(39) is available from Eq.(38).

5.2.3 The Multilayer Perceptron

As supervised classifiers, neural networks have recently become popular and have been used for speaker identification [6, 7, 9, 11, 13, 46, 47, 54]. Neural networks learn complex mappings between inputs and outputs and are particularly useful when the underlying statistics of the considered task are not well understood. The *multilayer perceptron* (MLP) is a type of neural network that has grown popular over the past several years. The MLP can be applied to speaker recognition as follows. First, the feature vectors are gathered for all speakers in the population. For a reference group of K speakers, a target vector is designed so that the ith component of the target vector corresponds to all feature vectors belonging to the ith speaker $(i = 1, \cdots, K)$ and it is labeled as "one" and the components for the remaining speakers are labeled as "zero" in the target vector. Thus, the MLP can be trained in the supervised manner for speaker identification. After training, for a test vector denoted as $\vec{x}'_s$, the MLP produces an output vector $\vec{O}(\vec{x}'_s) = [O_1(\vec{x}'_s), \cdots, O_K(\vec{x}'_s)]$. Accordingly, for a sequence of test vectors denoted as $\{\vec{x}'_s\}_{s=1}^S$, the test utterance is identified with speaker k only if $k = \arg\max_{1 \leq i \leq K} \bar{O}_i$, where $\bar{O}_i = \frac{1}{S}\sum_{s=1}^S O_i(\vec{x}'_s); \;\; i = 1, \cdots, K$.

In the experiments reported in the paper, the three-layered fully connected MLP is used and the 2-fold *cross-validation* technique [50] is employed for finding an appropriate architecture of the MLP for the considered task. As a result, the numbers of neurons in the hidden layer cover from 32 to 36 which depend upon the dimension of chosen feature vectors. As usual, the number of neurons in the input layer is the dimension of a feature vector used as the input (In the experiments reported in this paper, the number of neurons in the input layer is either 20 or 24.) and the number of neurons in the output layer is the population of speakers in the system. (In the experiments reported in this paper, there are 20 neurons in the output layer.) The backpropagation algorithm is used for training the MLPs [55].

5.2.4 Gaussian Mixture Model

As a parametric model, the *Gaussian mixture model* (GMM) was more recently applied to text-dependent speaker identification [52] and has demonstrated excellent performance for short test utterances. The basic idea underlying the GMM method lies in that the distribution of feature vectors extracted from a person's speech is modeled by a GMM density. For a feature vector denoted as $\vec{x}'_s$, the mixture density is defined as

$$P(\vec{x}_s|\Omega_i) = \sum_{j=1}^{M} \alpha_j^{(i)} P_j^{(i)}(\vec{x}_s), \qquad i = 1, \cdots, K. \tag{41}$$

The density is a weighted linear combination of M component uni-modal Gaussian densities described in Eq.(6), $P_j^{(i)}(\vec{x}_s)$, each parameterized by a mean vector, $\vec{\mu}_j^{(i)}$, and covariance matrix, $\Sigma_j^{(i)}$. Collectively, the parameters of a speaker's density model are denoted as $\Omega_i = \{\alpha_j^{(i)}, \vec{\mu}_j^{(i)}, \Sigma_j^{(i)}\}$. In this paper, diagonal covariance matrices are used like the work in [52]. Given a sequence of feature vectors from a person's training speech, maximum likelihood estimates of the model parameters are obtained using the EM algorithm [15, 52]. For a reference group of K speakers $\mathcal{K} = \{1, \cdots, K\}$ represented by models $\Omega_1, \cdots, \Omega_K$, the objective is to find the speaker model which has the maximum of a posteriori probability for the input vectors, $\{\vec{x}'_s\}_{s=1}^{S}$. Using logarithms and the assumed independence between observations, the test utterance is identified with speaker k only if $k = \arg\max_{1 \le i \le K} \sum_{s=1}^{S} \log P(\vec{x}'_s|\Omega_i)$ in which each $P(\vec{x}'_s|\Omega_i)$ $(i = 1, \cdots, K)$ is given in Eq.(41). In the experiments reported in the paper, 32 components $(M = 32)$ is used in the mixture model described in Eq.(41).

5.3 Results of Individual Classifiers and The Optimal Correspondency

First of all, we apply four chosen benchmark classifiers individually on each of four chosen common features for the text-independent speaker identification task. As mentioned above, the speech data in Set-1 were used for training each individual classifier. Since four feature sets are available from speech data in Set-1, each individual classifier was respectively trained on the four feature sets. As a result, each individual classifier will have four results corresponding to four feature sets when the speech data in Set-3 were used for testing. In the experiments, several thresholds are used to reject uncertain results and the best identifying result is obtained by trial and test. In this paper, the *best identifying result* yielded by a classifier or a method of combining multiple classifiers on a fixed test set is defined as the one with the *maximal identifying rate*. After four classifiers are respectively trained on four feature sets of speech data in Set-1, the best identifying results of four individual classifiers by testing speech data in Set-3 are shown in Table 1–4.

For convenience, we abbreviate names of combination methods described in this paper as follows. The linear combination method described in section 2 is called LIN-COM-DIF and the modified associative switch presented in section 3 is called M-ASSOC-SW. As for the methods described in section 4, the combination methods based upon Bayesian theory, Dempster-Shafer evidence theory and the voting principle are called BAYES, D-S and VOTING, respectively. In order to combine multiple classifiers with different features using LIN-COM-DIF and M-ASSOC-SW, the output of each classifier needs transforming into the standard form described in Eq.(1) using Eq.(2). As a result, two functions are chosen for the purpose as follows. The outputs of the MLP classifier and the GMM classifier are processed by the function $g(s) = s$ and the outputs of the distance classifier and the VQ classifier are processed by the function $g(s) = 1/s$. In the experiments of the M-ASSOC-SW method, the architecture of the MLP used for selecting a winner is a three-layered fully connected neural network with 4 input neurons, 4 hidden neurons and 4 output neurons and the standard backpropagation algorithm was used to train the MLP. Except the VOTING method, in the experiments, the speech data in Set-2 is employed to train each combination scheme or provides the a priori knowledge for combination, i.e. the training of each subscheme in the LIN-COM-DIF and the winner-take-all combination mechanism in the M-ASSOC-SW as well as the achievement of the confusion matrices in the BAYES and performance of each individual classifier (recognition rate and substitution rate) used in the D-S. In the VOTING method, the combination is directly fulfilled in the test set. (Actually, the test results on Set-2 may be helpful to the selection of an appropriate rejection rate during testing on Set-3). In the sequel, all results reported were obtained by testing speech data in Set-3.

In the current problem of combination, there are four different classifiers which were respectively trained on four different feature sets. It results in 16 possible cases for combination. We call each of such cases *correspondency* defined as the corresponding relation between those combined classifiers and their input features for a specific combination. For instance, K individual classifiers, *classifier-1,$\cdots$,classifier-K*, are respectively trained on K feature sets, *feature-1,$\cdots$,feature-K*. If these K classifiers are combined somehow after the training of K individual classifiers finishes, such a corresponding relation between K classifiers and their input features, i.e. *classifier-k* with *feature-k* $(k = 1, \cdots, K)$, will be called a correspondency. Moreover, the correspondency which can yields the best identifying result is called *optimal correspondency*. We have exhaustively investigated all 16 correspondencies using all combination methods described in section 2-4. For each combination method, 10 rejection thresholds which are uniformly distributed over the appropriate intervals defined in decision rules in Eq.(7), Eq.(23), Eq.(29), Eq.(32) and Eq.(34) were also selected to find the best identifying result by trial and test. As a result, We found an optimal correspondency from 16 possible correspondencies on the current speech database and the optimal correspondency is listed in Table 5. Using the optimal correspondency, we found that all combination methods described in the paper could yield the best identifying results. As a result, the best identifying results produced by different combination methods on the optimal correspondency and the corresponding rejection thresholds are shown in Table 6.

For the purpose of comparison, we have done the experiments on the optimal correspondency using the original associative switch in [69] and the proposed method in section 3. In [69], several methods were proposed for producing the desired output of the MLP used for selecting a winner. In the experiments, we exhaustively used those methods in [69] and only the best identifying result is reported here. We call the original associative switch ASSOC-SW and the best identifying results produced by the M-ASSOC-SW and ASSOC-SW are shown in Table 7. It is evident from the simulation that the modified associative switch outperforms the original one.

5.4 The Results for Comparison

As mentioned above, there are 16 possible correspondencies. In addition, there are several methods of combining multiple classifiers available in this paper. Although we have exhaustively done experiments on all correspondencies, we cannot report all of experimental results due to the limited space. For the purpose of comparison, we merely report some typical experiments for exploring different combinations. It should still be noted that several thresholds have been used in combinations of classifiers and only best identifying results are reported here.

Since there are four results of each classifier on four features of the same raw data, it is natural to consider such a correspondency to use the features such that individual classifiers can respectively achieve the best identifying result. According to results reported in Table 1–4, we could achieve the correspondency listed in Table 8. For convenience, the correspondency between classifiers and features is called correspondency-1. Accordingly, the results of correspondency-1 using different combination methods are shown in Table 9. However, its performance is slightly worse than the optimal correspondency's.

To investigate the complementarity among different features, the experiments of combining classifiers (the same type) with four different features have been conducted. In the experiments, one kind of classifier is chosen and respectively trained with four different features of raw data. Due to the limited space, in Table 11, we merely report the results of so-called correspondency-2 listed in Table 10. For other similar correspondencies, the results of other kinds of classifiers chosen in this paper are quite similar to the results on correspondency-2.

We have also conducted some experiments for combining four different classifiers with the same feature. The circumstance often occurs in most of pattern recognition problems. For convenience, correspondency-3, correspondency-4 and correspondency-5 denote correspondencies listed in Table 12–14, respectively. In addition, it was difficult to obtain better results than the best one of the individual classifier reported in Table 4 (the GMM classifier with the feature MEL-CEP) by combining four classifier with the feature CEPS. Therefore, the results of combining classifiers with the feature CEPS is not reported here. In addition, it is worth noting that in the circumstance of multiple classifiers with the same feature the LIN-COM-DIF method is degenerated into the method in [65]. That is, it is just the case that there is the unique β_1 and the value β_1 is always one in Eq.(4). Here, we still call the method LIN-COM-DIF for consistency. Accordingly, the results of correspondency-3, correspondency-4 and correspondency-5 are shown in Table 15–17, respectively.

On the basis of all experimental results, it is evident that the proposed method called LIN-COM-DIF could achieve the improved results for all cases. The method called BAYES could also achieve the satisfactory results for all cases except the correspondency-3, which is consistent with other applications of the BAYES method [50, 68].

6 Conclusions

We have described several methods of combining multiple classifiers with different features and their application to text-independent speaker identification In particular, we classify the state-of-the-art techniques for combining multiple classifiers into three frameworks. The methods in the same framework share the similar principle for combination. Based upon the experimental results, we have demonstrated that the performance of the text-independent speaker identification system is significantly improved and the methods of combining multiple classifiers with different features described in this paper outperform not only the individual classifiers but also the methods of combining multiple classifiers with the same feature. Moreover, it is evident from simulations that the proposed linear combination method outperforms other methods described in the paper. However, there are two open problems in the combination of multiple classifiers with different features. One is that there is no analysis of value of information from dependent classifiers in the case of different features as input though the topic has been recently discussed in the case of the same feature as the input of dependent classifiers [35]. The other is that for a given task an effective method of searching for an optimal correspondency on available classifiers and different features will be needed to be developed though the exhaustive way might work as it did in the paper. We shall explore these problems in our ongoing research.

Acknowledgments

Authors are very grateful to anonymous referees for their useful comments. This work was partially supported by National Science Foundation of China with Grants 69571002 & 69475007, the Climbing Program – National Key Project for Fundamental Research in China with Grant NSC 92097 and Foundation for Chinese Young Scientists Returning from Overseas.

References

1. M. A. Abidi and R. C. Gonzalez. *Data Fusion in Robotics and Machine Intelligence*. Academic Press, San Diego, 1992.

2. C. Agnew. Multiple probability assessments by dependent experts. *J. Am. Stat. Assoc.*, 80:343–347, 1985.

3. B. S. Atal. Automatic speaker recognition based on pitch contour. *J. Acoust. Soc. Am.*, 52(6):1687–1697, 1972.

4. B. S. Atal. Effectiveness of linear prediction characteristics of the speech waves for automatic speaker identification and verification. *J. Acoust. Soc. Am.*, 55(6):1304–1312, 1974.

5. R. Battiti and A. M. Colla. Democracy in neural nets: voting schemes for classification. *Neural Networks*, 7(4):691–708, 1994.

6. Y. Bennani. A modular and hybrid connectionist system for speaker identification. *Neural Computation*, 7(4):791–798, 1995.

7. Y. Bennani, F. Fogelman, and P. Gallinari. A connectionist approach for speaker identification. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1990. 265–268.

8. Y. Bennani and P. Gallinari. On the use of TDNN extracted features information in talker identification. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1991. 385–388.

9. Y. Bennani and P. Gallinari. Connectionist approaches for automatic speaker recognition. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, 1994. 95–102.

10. S. Chatterjee and S. Chatterjee. On combining expert opinions. *Am. J. Math. Management Sci.*, 7(1):271–295, 1987.

11. K. Chen, D. Xie, and H. Chi. Speaker identification based on hierarchical mixture of experts. In *Proc. World Congress on Neural Networks*, Washington D.C., 1995. I493–I496.

12. K. Chen, D. Xie, and H. Chi. A modified HME architecture for text-dependent speaker identification. *IEEE Trans. Neural Networks*, 7(5):1309–1313, 1996.

13. K. Chen, D. Xie, and H. Chi. Speaker identification using time-delay HMEs. *Int. J. Neural Systems*, 7(1):29–43, 1996.

14. K. Chen, D. Xie, and H. Chi. Text-dependent speaker identification based on input/output HMMs: an empirical study. *Neural Processing Letters*, 3(2):81–89, 1996.

15. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. B*, 39:1–38, 1977.

16. J. P. Dickenson. Some statistical results in the combination of forecasts. *Oper. Res. Q.*, 24:253–260, 1973.

17. J. P. Dickenson. Some comments on the combination of forecasts. *Oper. Res. Q.*, 26:205–210, 1975.

18. G. Doddington. Speaker recognition – identifying people by their voice. *Proceedings of IEEE*, 73(11):1651–1664, 1986.

19. K. R. Farrell, R. J. Mammone, and K. T. Assaleh. Speaker recognition using neural network classifier. *IEEE Trans. Audio, Speech Processing*, 2(1):194–205, 1994.

20. S. French. Group consensus probability distributions: a critical survey. In J. M. Bernardo, D. V. Lindley M. H. DeGroot, and A. F. M. Smith, editors, *Bayesian Statistics*, volume 2. Elsevier Science Publishers, North-Holland, 1985.

21. S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans. Acoust. Speech, Signal Processing*, 29(2):254–272, 1981.

22. S. Furui. Comparison of speaker recognition methods using statistical features and dynamic features. *IEEE Trans. Acoust. Speech, Signal Processing*, 29(3):197–200, 1981.

23. S. Furui. Research on individual features in speech waves and automatic speaker recognition techniques. *Speech Communication*, 5(2):183–197, 1986.

24. S. Furui. An overview of speaker recognition technology. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, 1994. 1–9.

25. S. Furui, F. Itakura, and S. Saito. Talker recognition by longtime averaged speech spectrum. *Electronics and Communications in Japan*, 55-A(10):54–61, 1972.

26. A. Gelfand, B. Ballick, and D. Dey. Modeling expert opinion arising as a partial probabilistic specification. *J. Am. Stat. Assoc.*, 90:598–604, 1995.

27. C. Genest and J. V. Zidek. Combining probability distributions: a critique and an annotated bibliography. *Statist. Sci.*, 1:114–148, 1986.

28. H. Gish and M. Schmidt. Text-independent speaker identification. *IEEE Signal Processing Magazine*, pages 18–32, Oct. 1994.

29. H. Gish, M. Schmidt, and A Mieke. A robust segmental method for text-independent speaker identification. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1994. 145–148.

30. C. W. J. Granger and R. Ramanathan. Improved methods of combining forecasts. *J. Forecasting*, 3:197–204, 1984.

31. S. Grossberg. Competition, decision and consensus. *J. Mathematical Analysis and Applications*, 66:470–493, 1978.

32. J. He, L. Liu, and G. Palm. A text-independent speaker identification system based on neural networks. In *Proc. Int. conf. Spoken Language Processing*, Yokohama, 1994.

33. A. Higgins, L. Bahler, and J. Porter. Voice identification using nearest-neighbor distance measure. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1993. 375–378.

34. M. Homayounpour and G. Chollet. A comparison of some relevant parametric representations for speaker verification. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, 1994. 1–9.

35. R. A. Jocobs. Methods for combining experts' probability assessments. *Neural Computation*, 7(5):867–888, 1995.

36. R. A. Jocobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

37. T. Kohonen. *Self-organization and associative memory*. Springer-Verlag, Tokyo, 1988.

38. K. Li and E. Wrench. An approach to text-independent speaker recognition with short utterances. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1983. 555–558.

39. T. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantization. *IEEE Trans. Communications*, 28(1):84–95, 1980.

40. D. V. Lindley. Reconciliation of discrete probability distributions. In J. M. Bernardo, D. V. Lindley M. H. DeGroot, and A. F. M. Smith, editors, *Bayesian Statistics*, volume 2. Elsevier Science Publishers, North-Holland, 1985.

41. E. J. Mandler and J. Schurmann. Combining the classification results of independent classifiers based on the Dempster/Shafer theory of evidence. *Pattern Recognition and Artificial Intelligence*, 5:381–393, 1988.

42. J. Markel, B. Oshika, and A. Gray Jr. Text-independent speaker recognition from a large linguistically unconstrained time-space data base. *IEEE Trans. Acoust., Speech, Sigal Processing*, 27:74–82, 1979.

43. T. Matsui and S. Furui. A text-dependent speaker recognition method robust against utterance variations. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1991. 377–380.

44. T. Matsui and S. Furui. Speaker recognition technology. *NTT Review*, 7(2):40–48, 1995.

45. S. J. Nowlan. Competing experts: An experimental investigation of associative mixture models. Tech. Rep. CRG-TR-90-5, Department of Computer Science, University of Toronto, 1990.

46. J. Oglesby and J. S. Mason. Optimization of neural models for speaker identification. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1990. 261–264.

47. J. Oglesby and J. S. Mason. Radial basis function networks for speaker recognition. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1991. 393–396.

48. J. P. Openshaw, Z. P. Sun, and J. S. Mason. A comparison of composite features under degraded speech in speaker recognition. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1993. II371–II374.

49. D. O'Shaughnessy. Speaker recognition. *IEEE ASSP Magazine*, 3(4):4–17, 1986.

50. M. P. Perrone. Improving regression estimation: averaging methods of variance reduction with extensions to general convex measure optimization. Ph.D. thesis, Department of Physics, Brown University, 1993.

51. L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, Prentice-Hall, New Jersey, 1993.

52. D. A. Reynolds. A Gaussian mixture modeling approach to text-independent speaker identification. Ph.D. Thesis, Department of Electrical Engineering, Georgia Institute of Technology, 1992.

53. G. Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

54. L. Rudasi and S. A. Zahorian. Text-independent talker identification with neural networks. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1991. 389–392.

55. D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing: Explorations in the microstructure of cognition*. MIT Press, Cambridge, 1986.

56. M. R. Sambur. Selection of acoustic features for speaker identification. *IEEE Trans. Acoust., Speech, Signal Processing*, 23:176–182, 1975.

57. R. Schwartz, S. Roucos, and M. Berouti. The application of probability density estimation to text-independent speaker identification. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1982. 1649–1652.

58. G. Shafer. *A Mathematical Theory of Evidence*. MIT Press, Cambridge, 1976.

59. F. Soong, A. Rosenberg, L. Rabiner, and B. Juang. A vector quantization approach to speaker recognition. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1985. 387–390.

60. F. K. Soong and A. E. Rosenberg. On the use of instantaneous and transitional spectral information in speaker recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(6):871–879, 1988.

61. C. Y. Suen, T. A. Nadal, T. A. Mai, R. Legault, and L. Lam. Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts. In C. Y. Suen, editor, *Forniers in Handwriting Recognition*, Montreal: Concordia University, 1990. 131–143.

62. A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust., Speech, Signal Processing*, 37:328–339, 1989.

63. R. Wohlford, E. Wrench Jr., and B. Lamdel. A comparison of four techniques for automatic speaker recognition. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1980. 908–911.

64. J. Wolf. Efficient acoustic parameters for speaker recognition. *J. Acoust. Soc. Am.*, 51(6):2044–2056, 1972.

65. L. Xu and M. I. Jordan. EM learning on a generalized finite mixture model for combining multiple classifiers. In *Proc. World Congress on Neural Networks*, San Diego, 1993. IV227–IV230.

66. L. Xu and M. I. Jordan. A modified gating network for the mixtures of experts architecture. In *Proc. World Congress on Neural Networks*, San Diego, 1994. II405–II410.

67. L. Xu, M. I. Jordan, and G. E. Hinton. An alternative model for mixture of experts. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, pages 633–640, Cambridge MA, 1995. MIT Press.

68. L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Sys. Man. Cybern.*, 23(3):418–435, 1992.

69. L. Xu, A. Krzyzak, and C. Y. Suen. Associative switch for combining multiple classifiers. *Journal of Artificial Neural Networks*, 1(1):77–100, 1994.

70. L. Xu, J. Oglesby, and J. S. Mason. The optimization of perceptually-based features for speaker identification. In *Proc. Int. conf. Acoust., Speech, Signal Processing*, 1989. 520–523.

71. Z. Zhang, I. Harmann, J. Guo, and R. Suchenwirth. A recognition method of printed Chinese character by feature combination. *Int. J. Research and Engineering – Postal Applications*, 1:77–82, 1989.

Table 1: The results(%) of the distance classifier

Feature (Input)	Identification	Substitution	Rejection	Reliability
LPCC	74.46	18.19	7.35	80.37
LPC-CEP	74.80	13.38	11.82	84.83
CEPS	64.66	25.50	9.84	71.72
MEL-CEP	76.03	16.46	7.51	82.20

Table 2: The results(%) of the vector quantization classifier

Feature (Input)	Identification	Substitution	Rejection	Reliability
LPCC	73.33	14.44	13.23	83.36
LPC-CEP	88.30	3.20	8.50	96.51
CEPS	80.87	16.50	2.63	83.05
MEL-CEP	88.49	1.30	10.21	98.55

Table 3: The results(%) of the multilayer perceptron classifier

Feature (Input)	Identification	Substitution	Rejection	Reliability
LPCC	88.57	6.93	4.50	92.74
LPC-CEP	86.93	8.16	4.91	91.43
CEPS	65.05	22.33	12.62	74.44
MEL-CEP	83.75	5.10	11.15	93.27

Table 4: The results(%) of the Gaussian mixture model classifier

Feature (Input)	Identification	Substitution	Rejection	Reliability
LPCC	86.32	6.97	6.71	92.53
LPC-CEP	91.65	2.82	5.53	97.02
CEPS	81.67	11.09	7.24	88.04
MEL-CEP	91.73	1.37	6.90	98.53

Table 5: The optimal correspondency on classifiers and features

Classifier	Distance-Classifier	VQ	MLP	GMM
Feature (Input)	LPCC	MEL-CEP	LPC-CEP	MEL-CEP

Table 6: The results(%) of the optimal correspondency using different combination methods

Method	Rejection-Threshold	Identification	Substitution	Rejection	Reliability
LIN-COM-DIF	0.1	97.33	0.00	2.67	100.0
M-ASSOC-SW	0.6	94.78	5.22	0.00	94.78
BAYES	0.9	97.07	0.00	2.93	100.0
D-S	0.6	96.38	1.64	1.98	98.33
VOTING	2	95.58	0.69	3.73	99.28

Table 7: The results(%) of the optimal correspondency using M-ASSOC-SW and ASSOC-SW

Method	Identification	Substitution	Rejection	Reliability
ASSOC-SW	93.25	6.75	0.00	93.25
M-ASSOC-SW	94.78	5.22	0.00	94.78

Table 8: Correspondency-1 on classifiers and features

Classifier	Distance-Classifier	VQ	MLP	GMM
Feature (Input)	MEL-CEP	MEL-CEP	LPCC	MEL-CEP

Table 9: The results(%) of correspondency-1 using different combination methods

Method	Identification	Substitution	Rejection	Reliability
LIN-COM-DIF	97.17	1.23	1.60	98.75
M-ASSOC-SW	93.87	4.69	1.45	95.25
BAYES	96.80	3.20	0.00	96.80
D-S	94.13	2.36	3.61	97.56
VOTING	93.75	0.91	5.34	99.04

Table 10: Correspondency-2 on classifiers and features

Classifier	MLP	MLP	MLP	MLP
Feature (Input)	LPCC	LPC-CEP	CEPS	MEL-CEP

Table 11: The results(%) of correspondency-2 using methods of combining classifiers

Method	Identification	Substitution	Rejection	Reliability
LIN-COM-DIF	93.80	6.12	2.08	95.79
M-ASSOC-SW	91.03	8.47	0.5	91.49
BAYES	91.99	7.89	0.12	92.10
D-S	89.38	6.93	3.69	92.73
VOTING	88.87	6.30	4.83	93.38

Table 12: Correspondency-3 on classifiers and the feature

Classifier	Distance-Classifier	VQ	MLP	GMM
Feature (Input)	LPCC	LPCC	LPCC	LPCC

Table 13: Correspondency-4 on classifiers and the feature

Classifier	Distance-Classifier	VQ	MLP	GMM
Feature (Input)	LPC-CEP	LPC-CEP	LPC-CEP	LPC-CEP

Table 14: Correspondency-5 on classifiers and the feature

Classifier	Distance-Classifier	VQ	MLP	GMM
Feature (Input)	MEL-CEP	MEL-CEP	MEL-CEP	MEL-CEP

Table 15: The results(%) of correspondency-3 using methods of combining multiple classifiers

Method	Identification	Substitution	Rejection	Reliability
LIN-COM-DIF	92.77	5.40	1.83	94.50
M-ASSOC-SW	89.22	8.46	2.32	91.34
BAYES	77.25	16.04	6.71	82.80
D-S	91.05	5.60	3.35	94.21
VOTING	90.58	4.62	4.80	93.47

Table 16: The results(%) of correspondency-4 using methods of combining multiple classifiers

Method	Identification	Substitution	Rejection	Reliability
LIN-COM-DIF	95.29	3.53	1.18	96.43
M-ASSOC-SW	91.86	6.46	1.68	93.43
BAYES	96.11	3.89	0.00	96.11
D-S	91.27	6.59	2.14	93.27
VOTING	92.98	2.78	4.24	97.10

Table 17: The results(%) of correspondency-5 using methods of combining multiple classifiers

Method	Identification	Substitution	Rejection	Reliability
LIN-COM-DIF	96.50	2.54	1.96	98.43
M-ASSOC-SW	93.15	6.85	0.00	93.15
BAYES	95.20	4.76	0.04	95.23
D-S	94.13	4.80	1.07	95.15
VOTING	91.12	0.91	7.97	99.01