ELSEVIER

Neurocomputing 20 (1998) 227–252

NEUROCOMPUTING

A method of combining multiple probabilistic classifiers through soft competition on different feature sets

Ke Chen[a,b,*], Huisheng Chi[a]

[a] *National Laboratory of Machine Perception and Center for Information Science, Peking University, Beijing 100871, China*

[b] *Department of Computer and Information Science and Center for Cognitive Science, The Ohio State University, Columbus, OH 43210-1277, USA*

Received 12 January 1997; accepted 23 March 1998

Abstract

A novel method is proposed for combining multiple probabilistic classifiers on different feature sets. In order to achieve the improved classification performance, a generalized finite mixture model is proposed as a linear combination scheme and implemented based on radial basis function networks. In the linear combination scheme, soft competition on different feature sets is adopted as an automatic feature rank mechanism so that different feature sets can be always simultaneously used in an optimal way to determine linear combination weights. For training the linear combination scheme, a learning algorithm is developed based on Expectation–Maximization (EM) algorithm. The proposed method has been applied to a typical real-world problem, viz., speaker identification, in which different feature sets often need consideration simultaneously for robustness. Simulation results show that the proposed method yields good performance in speaker identification.

Keywords: Combination of multiple classifiers; Soft competition; Different feature sets; Expectation–maximization (EM) algorithm; Speaker identification

1. Introduction

The problem of pattern classification can be stated as follows: Given a set of training data, each with an associated label, find a classification system that will

* Corresponding author. E-mail: chen@cis.pku.edu.cn

0925-2312/98/$ – see front matter
PII S0925-2312(98)00019-8

produce the correct label for any data drawn from the same source as the training data. As illustrated in Fig. 1a, in general, such a classification system is composed of three stages: preprocessing, feature extraction, and classification. In particular, feature extraction is necessary to avoid a so-called curse of dimensionality problem [18] which may lead to prohibitively expensive computation in the stage of classification. Therefore, the performance of a classification system highly depends upon a feature set used. For a classification task, numerous types of features can be extracted from the same raw data by means of different methods. A selection technique is often adopted to find an optimal feature set for use in classification [6]. Sometimes, however, it is impossible to find such an optimal feature set. Instead several different feature sets can result in similar classification performance so that none of them can be optimal or robust for a specific classification task. Because different feature sets represent raw data from different viewpoints, the simultaneous use of different feature sets can lead to a better or robust classification result. As illustrated in Fig. 1b, this kind of problems are called *pattern classification on different feature sets* in this paper. There are many real-world problems belonging to this category. A typical example is *speaker identification* that classifies an unlabeled voice token as belonging to one of reference speakers. For this problem, several different spectrum feature sets have been turned out useful, but none of them can be regarded as an optimal or robust one. It has been suggested that multiple spectrum feature sets need consideration simultaneously for robustness in speaker identification [20,21]. As a result, a technique that

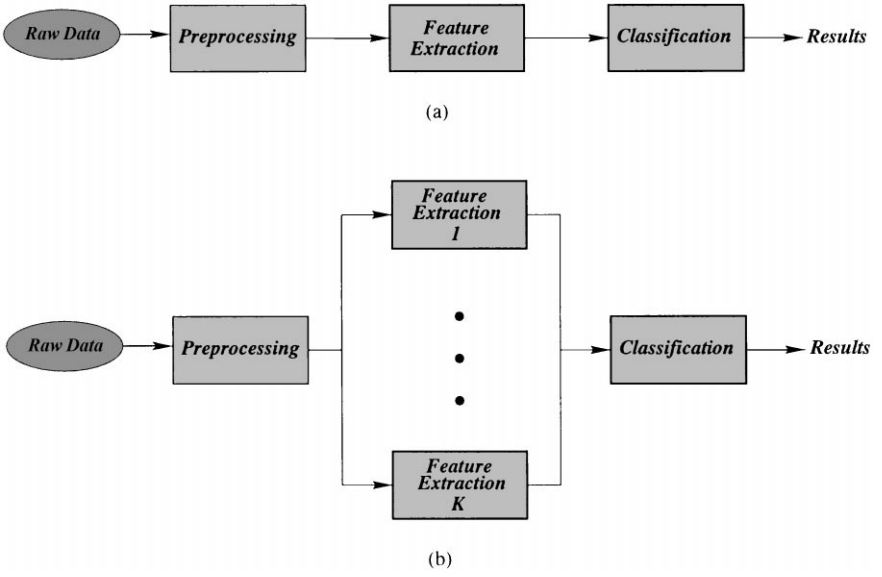


Fig. 1. Pattern recognition systems. (a) The general schematic structure of a pattern recognition system based on a robust feature set. (b) The schematic structure of a pattern recognition system based on different feature sets.

efficiently utilizes different feature sets becomes a solution to pattern classification on different feature sets.

For simultaneous use of different feature sets, a traditional method is to lump different feature vectors together into a single composite feature vector. Although there are several methods to form a composite feature vector, the use of a composite feature set may result in the following problems: (1) Curse of dimensionality; the dimension of a composite feature vector becomes much higher than any of component feature vectors. (2) Difficulty in formation; it is often difficult to lump several different feature vectors together due to their diversified forms. (3) Redundancy; the component feature vectors are usually not independent of each other. Therefore, the composite-feature based method can simply achieve limited success. Recently, combination of multiple classifiers has been viewed as a new direction for the development of highly reliable pattern recognition systems. Preliminary results indicate that combination of several complementary classifiers may lead to the improved performance [26,25,3,38,33,32,35,23,46]. There are at least two reasons for necessity of combining multiple classifiers. On the one hand, there are a number of classification algorithms developed from different theories and methodologies in almost all the current pattern recognition application areas. Each of these classifiers may reach a certain degree of success, but none of them is totally perfect or so good as expected in practical applications. On the other hand, the demand for a solution to pattern classification on different feature sets becomes the other reason to do so. A combination technique allows multiple classifiers to work on different feature sets so that different feature sets can be utilized simultaneously. Therefore, a method of combining multiple classifiers on different feature sets provides an alternative way to solve the problem of pattern classification on different feature sets.

From the viewpoint of statistics, combination of multiple classifiers can be viewed as combination of multiple probability distributions if each classifier is interpreted as an estimator of probability distribution. In general, there are two frameworks to perform such a combination [27]. One is that of a decision maker who consults several classifiers regarding some events. The classifiers express their opinions in the form of probability distributions. The decision maker must aggregate the classifiers' distributions into a single distribution that can be used to make the final decision. The other is the framework of linear opinion pools in which the decision maker forms a linear combination of classifiers' opinions. Under the two frameworks, there have been extensive studies in combination of multiple probabilistic classifiers [27,22,44,46,10,1]. In terms of pattern classification on different feature sets, combination techniques under the framework of a decision maker, e.g. a supra Bayesian procedure [27,46], can be directly used for this problem since outputs of the classifiers are merely considered for combination regardless of their inputs. On the other hand, several combination techniques under the framework of linear opinion pools can be also directly applied for combination of multiple classifiers on different feature sets [33,32,35,23,7,43]. These techniques are based on constrained or unconstrained least-squares regression with model selection to make a system achieve good generalization properties. However, the existing techniques of linear opinion pools with weights as veridical probabilities [27,45,44] cannot be used to deal with a problem of pattern

classification on different feature sets unless a single composite feature set is used since the process of generating weights used for linear combination highly depends upon inputs of multiple classifiers.

In this paper, we propose a new linear combination scheme to extend the existing techniques of linear opinion pools with weights as veridical probabilities for pattern classification on different feature sets. In contrast to the winner-take-all mechanism, *soft competition* is a concept that a competitor and its rivals can work for the same task together, but the winner plays a more important role than losers. In the linear combination scheme, we adopt such a soft competition mechanism on different feature sets to determine weights in an optimal way for linear combination. An EM learning algorithm is also proposed for parameter estimation in the linear combination scheme. To demonstrate its effectiveness, we have applied the proposed method to a real-world problem, viz., *speaker identification*, in which diversified feature sets need consideration simultaneously for robustness. Simulation results show that the proposed method yields satisfactory performance in speaker identification.

The remainder of this paper is organized as follows. Section 2 presents the methodology on the linear combination scheme through soft competition. Section 3 describes an EM learning algorithm for parameter estimation in the linear combination scheme. Section 4 reports simulation results on speaker identification. Conclusions are drawn in the final section.

2. Methodology

In this section, we first give the basic idea underlying the proposed linear combination scheme. Then we present a generalized finite mixture model as the linear combination scheme and give its implementation based on radial basis function networks.

2.1. Soft competition on different feature sets

For pattern classification on different feature sets, we assume that there are K ($K > 1$) different feature extraction methods so that K different feature sets can achieved from a raw data set. For an input sample $D^{(t)}$ in the raw data set, $\mathscr{X} = \{D^{(t)}, \boldsymbol{y}^{(t)}\}_{t=1}^{\mathrm{T}}$, therefore, K different feature vectors, $\boldsymbol{x}_1(D^{(t)}), \ldots, \boldsymbol{x}_K(D^{(t)})$, can be extracted from the sample $D^{(t)}$. To simplify the presentation, hereinafter, we drop the specific sample term, $D^{(t)}$, from these different feature vectors as $\boldsymbol{x}_1^{(t)}, \ldots, \boldsymbol{x}_K^{(t)}$.

Suppose that there is an optimal feature vector among those different feature vectors to represent the corresponding raw datum. Thus, a problem can be raised: which one is the optimal feature vector of the sample, $D^{(t)}$, among its K different feature vectors, $\boldsymbol{x}_1^{(t)}, \ldots, \boldsymbol{x}_K^{(t)}$? Apparently, a feature selection technique should be a solution to the problem. As pointed out in introduction, unfortunately, such a method is often not available in many real world problems. Here, we attempt to present an alternative solution to the problem. Prior to addressing the solution, we first introduce a set of binary indicator variables to represent the optimal feature vector. An

indicator, $I_k^{(t)}$, for feature vector $\boldsymbol{x}_k^{(t)}$ is defined as $I_k^{(t)} = 1$ if $\boldsymbol{x}_k^{(t)}$ is the optimal feature vector; otherwise, $I_k^{(t)} = 0$. According to the optimal feature definition, $\sum_{k=1}^{K} I_k^{(t)} = 1$ is always guaranteed. If we always use such an optimal feature vector to represent a raw datum and ignore the other feature vectors, there will exist a probabilistic relation between the raw datum and its optimal feature vector via. the indicator as follows:

$$P(\boldsymbol{x}_k^{(t)}) = P(D^{(t)}|I_k^{(t)} = 1). \tag{1}$$

Obviously, a solution to the aforementioned problem would be always available if such indicators were known. In practice, however, the indicators remain unknown or are typically missing data. It is more likely that there is no unique feature highly superior to other features for representing all the input samples. Therefore, the basic idea is to jointly use all the achieved feature vectors to represent a raw datum via. indicator variables. For doing so, we specify a finite mixture model as

$$P(D^{(t)}) = \sum_{k=1}^{K} P(D^{(t)}|I_k^{(t)} = 1)P(I_k^{(t)} = 1). \tag{2}$$

This mixture model provides an optimal way to utilize different feature sets through soft competition. In Eq. (2), those probability terms, $P(I_k^{(t)} = 1)$, will be used to determine the winner or losers. Obviously, the open problems are how to utilize the mixture model to combine multiple classifiers on different feature sets. In the sequel, we shall propose a generalized finite mixture model based on Eq. (2) and give its implementation to solve the problem.

2.2. Linear combination scheme

Consider a task of pattern classification on different feature sets, we assume that $N(N \geqslant K)$ probabilistic classifiers, $e_1, \ldots, e_N$, are employed to learn the classification task on a training set, respectively, in which the input of classifier e_j is the feature vector $\boldsymbol{x}_{k_j}$ $(j = 1, \ldots, N; 1 \leqslant k_j \leqslant K)$. Note that we simply consider a single sample here and, therefore, we drop its order t for simplicity. Suppose that the task is an M-category classification with class labels, $C_1, \ldots, C_M$. For a probabilistic classifier, e_j, its output vector, $\boldsymbol{p}_j(\boldsymbol{x}_{k_j})$, must satisfy the following conditions:

$$\boldsymbol{p}_j(\boldsymbol{x}_{k_j}) = [p_{j1}(\boldsymbol{x}_{k_j}), \ldots, p_{jM}(\boldsymbol{x}_{k_j})]^{\mathrm{T}}, \quad p_{jm}(\boldsymbol{x}_{k_j}) \geq 0 \text{ and } \sum_{m=1}^{M} p_{jm}(\boldsymbol{x}_{k_j}) = 1.$$

Here, $p_{jm}(\boldsymbol{x}_{k_j})$ denotes the probability that the sample D belongs to class C_m recognized by classifier e_j in terms of input feature vector $\boldsymbol{x}_{k_j}$. Note that we use $\boldsymbol{x}_{k_j}$ in the output vector to emphasize classifier e_j's input. The direct instances of probabilistic classifiers include those based on parametric or nonparametric density estimation, while other kinds of classifiers can be transformed into such classifiers, e.g. distance classifiers and neural network classifiers. Given the output vector of

classifier e_j, $\boldsymbol{u}_j(\boldsymbol{x}_{k_j}) = [u_{j1}(\boldsymbol{x}_{k_j}), \ldots, u_{jM}(\boldsymbol{x}_{k_j})]^{\mathrm{T}}$ that does not satisfy $u_{jm}(\boldsymbol{x}_{k_j}) \geq 0$ and $\sum_{m=1}^{M} u_{jm}(\boldsymbol{x}_{k_j}) = 1$, a transformation is defined as

$$p_{jm} = \frac{f(u_{jm})}{\sum_{m=1}^{M} f(u_{jm})}, \tag{3}$$

where $f(u_{jm}) \geq 0$ for $m = 1, \ldots, M$. There are various forms of the function $f(\cdot)$ used in Eq. (3) such as $f(u_{jm}) = u_{jm}$, $f(u_{jm}) = 1/u_{jm}$, and $f(u_{jm}) = \mathrm{e}^{-u_{jm}}$ when $u_{jm} \geq 0$ $(m = 1, \ldots, M)$ or $f(u_{jm}) = u_{jm}^2$, $f(u_{jm}) = 1/u_{jm}^2$, and $f(u_{jm}) = \mathrm{e}^{-u_{jm}^2}$ when $u_{jm} \leq 0$ $(m = 1, \ldots, M)$. Note that we drop the input feature vector, $\boldsymbol{x}_{k_j}$, in Eq. (3) for simplicity.

For an input–output pair $\{\boldsymbol{x}_{k_j}, \boldsymbol{y}\}$, where $\boldsymbol{y} = [y_1, \ldots, y_M]^{\mathrm{T}}$, $y_m \in \{0,1\}$, and $\sum_{m=1}^{M} y_m = 1$, classifier e_j in terms of input $\boldsymbol{x}_{k_j}$ specifies a distribution as

$$P(\boldsymbol{y}|\boldsymbol{x}_{k_j}, \Theta_j) = \prod_{m=1}^{M} [p_{jm}(\boldsymbol{x}_{k_j}|\Theta_j)]^{y_m}, \tag{4}$$

where Θ_j is the parameter vector of classifier e_j and has been already fixed as a constant vector after the classifier is trained. For a fixed $\boldsymbol{x}_{k_j}$, it is reduced to the multinomial distribution, while we can achieve a distribution specified by one of $p_{jm}(\boldsymbol{x}_{k_j}|\Theta_j)$'s for a fixed $\boldsymbol{y}$. Moreover, we assume that there are priors, $\Phi = \{(\alpha_{kj}(\boldsymbol{x}_k), \beta_k) \mid k = 1, \ldots, K;\ j = 1, \ldots, N\}$, for all the classifiers, where $\alpha_{kj}(\boldsymbol{x}_k)$ is associated with $P(\boldsymbol{x}_k)$ in Eq. (1) and β_k is an equivalent of $P(I_k = 1)$ in Eq. (2). For a generic input–output pair $\{D, \boldsymbol{y}\}$, we define a generalized finite mixture distribution on the basis of those priors in Φ as

$$\begin{aligned} P(\boldsymbol{y}|D, \Phi) &= \sum_{j=1}^{N} \left[\sum_{k=1}^{K} \beta_k \alpha_{kj}(\boldsymbol{x}_k) \right] P(\boldsymbol{y}|\boldsymbol{x}_{k_j}, \Theta_j) \\ &= \sum_{j=1}^{N} \sum_{k=1}^{K} \beta_k \alpha_{kj}(\boldsymbol{x}_k) \prod_{m=1}^{M} [p_{jm}(\boldsymbol{x}_{k_j}|\Theta_j)]^{y_m}, \end{aligned} \tag{5}$$

where $\alpha_{kj}(\boldsymbol{x}_k) \geq 0$, $\sum_{j=1}^{N} \alpha_{kj}(\boldsymbol{x}_k) = 1$, $\beta_k \geq 0$, and $\sum_{k=1}^{K} \beta_k = 1$. The generalized finite mixture model suggests a new scheme to combine multiple classifiers on different feature sets. As illustrated in Fig. 2, there are K subschemes in the linear combination scheme, where the kth subscheme performs estimation of the weights for combination of multiple classifiers in terms of the feature vector $\boldsymbol{x}_k$. In detail, $\alpha_{kj}(\boldsymbol{x}_k)$ is the weight produced by the kth subscheme for classifier e_j for linear combination in terms of the feature vector $\boldsymbol{x}_k$, while β_k is the probability that the kth subscheme with input $\boldsymbol{x}_k$ is used to produce the weights for linear combination. Here, we emphasize that those $\beta_k (k = 1, \ldots, K)$ play the role of soft competition on K different feature sets. Since maximum-likelihood estimation will be applied for learning in the next section, the linear combination scheme provides an optimal way to simultaneously use different feature sets for pattern classification.

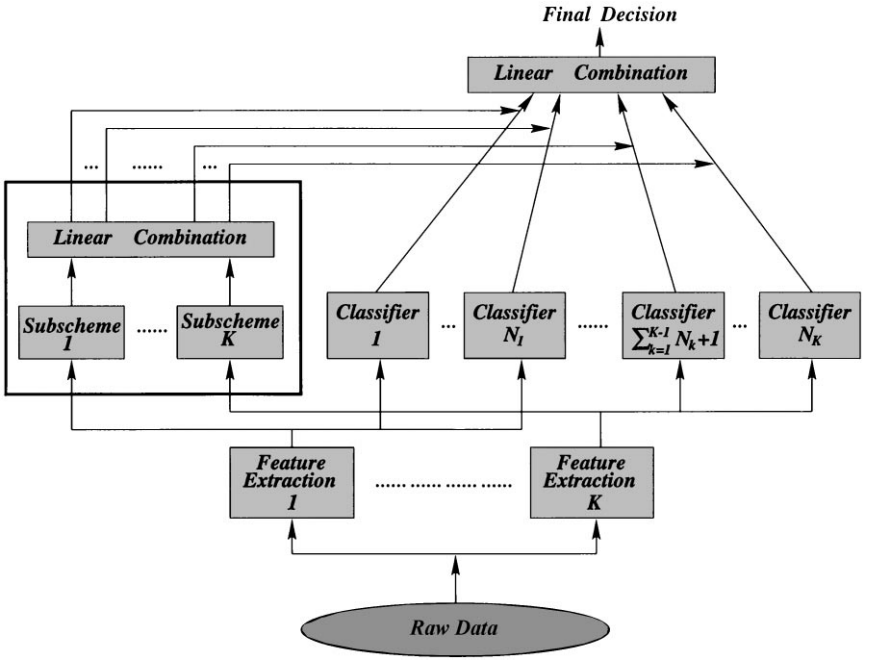


Fig. 2. The architecture of our linear combination scheme for pattern classification on different feature sets. Assume that K different feature sets are extracted from the same raw data set and N classifiers are employed where the kth feature set is used to train N_k different classifiers ($N_k \geq 1$ and $\sum_{k=1}^{K} N_k = N$). K subschemes on different feature sets in the large box are used to produce weights, $\alpha_{kj}(\boldsymbol{x}_k)$ and β_k ($j = 1, \ldots, N$; $k = 1, \ldots, K$), for linear combination of N classifiers on different feature sets.

In Eq. (5), all the priors $\alpha_{kj}(\boldsymbol{x}_k) \in \Phi\,(k = 1, \ldots, K;\ j = 1, \ldots, N)$ are conditional distributions on corresponding inputs $\boldsymbol{x}_k\,(k = 1, \ldots, K)$, respectively. As suggested by Xu et al. [45], in general, the parametric form of $\alpha_{kj}(\boldsymbol{x}_k)$ is defined as

$$\alpha_{kj}(\boldsymbol{x}_k, \lambda_{kj}, \Psi_{kj}) = \frac{\lambda_{kj}\Omega(\boldsymbol{x}_k, \Psi_{kj})}{\sum_{j=1}^{N} \lambda_{kj}\Omega(\boldsymbol{x}_k, \Psi_{kj})}, \tag{6}$$

where $\lambda_{kj} \geq 0$, $\sum_{j=1}^{N} \lambda_{kj} = 1$ for $k = 1, \ldots, K$, $j = 1, \ldots, N$. $\Omega(\boldsymbol{x}_k, \Psi_{kj})$ is a positive parametric function. In this paper, moreover, each $\Omega(\boldsymbol{x}_k, \Psi_{kj})$ is realized by a Gaussian distribution[1] as

$$\begin{aligned}\Omega(\boldsymbol{x}_k, \Psi_{kj}) &= \Omega(\boldsymbol{x}_k, \boldsymbol{m}_{kj}, \Sigma_{kj}) \\ &= \frac{1}{(2\pi)^{n/2}|\Sigma_{kj}|^{1/2}} \exp\left\{ -\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{m}_{kj})^{\mathrm{T}} \Sigma_{kj}^{-1} (\boldsymbol{x}_k - \boldsymbol{m}_{kj}) \right\},\end{aligned} \tag{7}$$

[1] When $\boldsymbol{x}_k$ cannot be modeled as a Gaussian distribution, a generalized linear model or a multilayer perceptron with a softmax output layer can be used as the parametric form of $\alpha_{kj}(\boldsymbol{x}_k)$. The EM learning algorithms have been also developed for parameter estimation in such models [30,15].

where n is the dimension of $\boldsymbol{x}_k$, and $\Psi_{kj} = (\boldsymbol{m}_{kj}, \Sigma_{kj})$ are parameters of the Gaussian distribution. The parametric form of $\alpha_{kj}(\boldsymbol{x}_k)$ can be viewed as a radial basis function network in which $\Omega(\boldsymbol{x}_k, \boldsymbol{m}_{kj}, \Sigma_{kj})$ is a basis function and those λ_{kj} are the corresponding weights. Thus, all the subschemes dependent on those priors can be implemented by a number of parametric functions or radial basis function networks. In the linear combination scheme, the information with respect to the outputs of classifiers, the desire labels, $\boldsymbol{y}$, and different inputs, $\boldsymbol{x}_k\,(k = 1, \ldots, K)$, is jointly considered for the final decision. Note that so far all the $\alpha_{kj}(\boldsymbol{x}_k)$ and $\beta_k\,(k = 1, \ldots, K; j = 1, \ldots, N)$ in Eq. (5) are still unknown. We shall propose a maximum-likelihood learning method for parameter estimation in the next section. Suppose that those priors have been already achieved, a linear combination can be defined based on Eq. (5) in the following manner. For an input sample D, $P(\boldsymbol{y}|D)$ is achieved by

$$P(y_m = 1|D,\Phi) = \sum_{j=1}^{N} \sum_{k=1}^{K} \beta_k \alpha_{kj}(\boldsymbol{x}_k) p_{jm}(\boldsymbol{x}_{k_j}), \quad m = 1, \ldots, M. \tag{8}$$

Using the linear combination, a decision rule is defined based on the maximum a posterior (MAP) principle as

$$m^* = \arg\max_{1 \le m \le M} P(y_m = 1|D,\Phi). \tag{9}$$

Thus, the sample D is classified as class m^*.

3. EM Algorithm for maximum-likelihood learning

In this section, we present a maximum-likelihood learning method for parameter estimation in the linear combination scheme on the basis of Expectation–Maximization (EM) algorithm [16].

To present the EM algorithm, we assume that N classifiers have been already trained on K ($K \leqslant N$) different feature sets extracted from the same training set, Given a cross-validation set, $\mathscr{X} = \{(D^{(t)}, \boldsymbol{y}^{(t)})\}_{t=1}^{\mathrm{T}}$, K different feature sets, $\{\boldsymbol{x}_1^{(t)}\}_{t=1}^{\mathrm{T}}, \ldots, \{\boldsymbol{x}_K^{(t)}\}_{t=1}^{\mathrm{T}}$, can be extracted from the input data set $\{D^{(t)}\}_{t=1}^{\mathrm{T}}$ with the same feature extraction methods. Moreover, all the samples in $\mathscr{X}$ are independent. Using the K different feature sets, $\{\boldsymbol{x}_k^{(t)}\}_{t=1}^{\mathrm{T}}\,(k = 1, \ldots, K)$, we train the linear combination scheme. For maximum-likelihood learning, therefore, the log-likelihood function is defined based on Eq. (5) as

$$\begin{aligned} L &= \log \prod_{t=1}^{\mathrm{T}} P(\boldsymbol{y}^{(t)}|D^{(t)},\Phi) \\ &= \sum_{t=1}^{\mathrm{T}} \log P(\boldsymbol{y}^{(t)}|D^{(t)},\Phi) \\ &= \sum_{t=1}^{\mathrm{T}} \log \left[\sum_{j=1}^{N} \sum_{k=1}^{K} \beta_k \alpha_{kj}(\boldsymbol{x}_k^{(t)}) P(\boldsymbol{y}^{(t)}|\boldsymbol{x}_{k_j}^{(t)}, \Theta_j) \right]. \end{aligned} \tag{10}$$

Note that $\boldsymbol{x}_{k_j}^{(t)}(1 \le k_j \le K)$ still stands for the tth input of classifier e_j. For the log-likelihood function, we apply an EM algorithm [16] for parameter estimation by introducing a set of indicators as missing data or unobserved data to observed data in $\mathscr{X}$. To facilitate the presentation, we skip the detailed derivation of the EM algorithm here and put it in appendix. As a result, the resulting EM algorithm is summarized as follows.

3.1. EM Algorithm for the linear combination scheme

1. **Initialization at $s = 0$**

For $j = 1, \ldots, N$ and $k = 1, \ldots, K$, set $\beta_k^{(s)} = 1/K$ and $\lambda_{kj}^{(0)} = 1/N$. Randomly initialize $\Psi_{k1}, \ldots, \Psi_{kN}$ so that $\Psi_{k1}^{(0)} = \cdots = \Psi_{kN}^{(0)}$ for $k = 1, \ldots, K$ subject to $\alpha_{kj}^{(0)}(\boldsymbol{x}_k) = 1/N$ for $j = 1, \ldots, N$.

2. **The EM procedure at $s > 0$**

2.1. *E-step*: For each pair $(D^{(t)}, \boldsymbol{y}^{(t)}) \in \mathscr{X}$, calculate the posterior probabilities: for $j = 1, \ldots, N$; $k = 1, \ldots, K$, the posterior probabilities, $h_k^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)})$ and $h_{kj}^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)})$, can be achieved by

$$h_k^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)}) = \frac{\beta_k^{(s)} \sum_{j=1}^{N} \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)}{\sum_{j=1}^{N} \sum_{k=1}^{K} \beta_k^{(s)} \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)} \tag{11}$$

and

$$h_{kj}^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)}) = \frac{\beta_k^{(s)} \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)}{\sum_{j=1}^{N} \sum_{k=1}^{K} \beta_k^{(s)} \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)}. \tag{12}$$

2.2. *M-step*:. Based on the current posterior probabilities achieved in the *E*-step, find a new estimate for each parameter: for $j = 1, \ldots, N$, $k = 1, \ldots, K$, those parameters are updated as

$$\boldsymbol{m}_{kj}^{(s+1)} = \frac{1}{\sum_{t=1}^{\mathrm{T}} h_{kj}^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)})} \sum_{t=1}^{\mathrm{T}} h_{kj}^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)}) \boldsymbol{x}_k^{(t)}, \tag{13}$$

$$\Sigma_{kj}^{(s+1)} = \frac{1}{\sum_{t=1}^{\mathrm{T}} h_{kj}^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)})} \sum_{t=1}^{\mathrm{T}} h_{kj}^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)}) [\boldsymbol{x}_k^{(t)} - \boldsymbol{m}_{kj}^{(s+1)}][\boldsymbol{x}_k^{(t)} - \boldsymbol{m}_{kj}^{(s+1)}]^{\mathrm{T}}, \tag{14}$$

$$\lambda_{kj}^{(s+1)} = \frac{1}{T} \sum_{t=1}^{\mathrm{T}} h_{kj}^{(s)}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_k^{(t)}), \tag{15}$$

$$\beta_k^{(s+1)} = \frac{1}{T} \sum_{t=1}^{\mathrm{T}} h_k^{(s)}(\boldsymbol{y}^{(t)} \mid D^{(t)}), \tag{16}$$

and

$$\alpha_{kj}^{(s+1)}(\boldsymbol{x}_k^{(t)}) = \frac{\lambda_{kj}^{(s+1)} \Omega(\boldsymbol{x}_k^{(t)}, \boldsymbol{m}_{kj}^{(s+1)}, \Sigma_{kj}^{(s+1)})}{\sum_{j=1}^{N} \lambda_{kj}^{(s+1)} \Omega(\boldsymbol{x}_k^{(t)}, \boldsymbol{m}_{kj}^{(s+1)}, \Sigma_{kj}^{(s+1)})}. \tag{17}$$

3. Repeat Step 2 until a pre-specified termination condition is satisfied.

Using the EM algorithm, all unknown parameters in Eq. (5) are learned based on the cross-validation set $\mathscr{X}$. After learning, both the linear combination described in Eq. (8) and the MAP decision rule defined in Eq. (9) can work together to combine multiple probabilistic classifiers on different feature sets.

4. Simulations

In this section, we demonstrate the effectiveness of the linear combination scheme as illustrated in Fig. 2. In order to evaluate its performance, we have applied our method to a real world problem called speaker identification.

Speaker identification is to classify an unlabeled voice token as belonging to one of a set of registered reference speakers. A speaker identification system can be either *text-dependent* or *text-independent*. By text-dependent, we mean that the text in both training and test is the same or known. In contrast, the text in a text-independent speaker identification system should be arbitrary in either training or test. Speaker identification is a rather hard task for learning since a person's voice always changes in time. In addition, many other factors, e.g. short-term sickness, emotion, fatigue, and the words spoken, may significantly alter personal voice characteristics [21]. Although there is a long history to explore speaker's feature, the unique robust feature has still not been discovered so far. Instead many kinds of spectral features have been reported to be useful to speaker identification [17,34,20,9,21], and there is no sophisticated feature selection technique. Different feature sets have been simultaneously considered for robustness in speaker identification. To use multiple features simultaneously, a traditional method is to lump two or more different feature vectors together as a single composite feature vector [20,21]. However, the performance of the composite-feature based system is not significantly improved. To some extent, the use of a composite feature set results in the curse of dimensionality problem. In particular, the problem will become quite serious when time-delay neural computation techniques are used [42,5,4,14]. As a result, speaker identification becomes a typical task of pattern classification on different feature sets. In the sequel, we shall report simulation results produced by our purposed method for both text-dependent and text-independent speaker identification, respectively. Moreover, comparative results will be also presented to show the effectiveness of our method from a different viewpoint.

4.1. Results on text-dependent speaker identification

We have applied our method to text-dependent speaker identification. In simulations, we chose isolated digits as the fixed text used in both training and test. The method has been extensively used to test a text-dependent speaker identification system [17,40,13]. The acoustic database consists of 10 isolated digits from '0' to '9' uttered in Chinese (Mandarin dialect). All the utterances were recorded in three different sessions and 20 male speakers were registered in the database. For each digit, 200 utterances (10 utterances/speaker) were recorded in each session. The technical

details of preprocessing are briefly summarized as follows, (1) 16-bit A/D-converter with 11.025 KHz sampling rate, (2) processing the data with a pre-emphasis filter $H(z) = 1 - 0.95z^{-1}$, and (3) 25.6 ms Hamming window with 12.8 ms overlapping for blocking an utterance into several feature frames for the short-time spectral analysis. In simulations, we adopted four common features used in text-dependent speaker identification, i.e. 19-order delta-cepstrum, 19-order LPC based cepstrum, 19-order Mel-scale cepstrum, and 15-order LPC coefficients [36].

The hierarchical mixture of experts (HME) is a modular neural network architecture recently proposed by Jordan and Jocobs [30]. Our earlier work showed that the HME architecture yielded better identification results in text-dependent speaker identification in contrast to conventional methods [12,13]. Based on our earlier work, we employed the HME architecture as an individual classifier. Therefore, 40 HMEs were used in simulations so that for each digit four two-level HMEs with 2–8 structure[2] were trained on the four different feature sets using the EM algorithm [30,15], respectively. For a digit, five utterances of each speaker recorded in the first session were merely used for training and all the utterances of this digit recorded in other two sessions were used for test. Tests on utterances recorded in the second and the third sessions are called TEST-1 and TEST-2, respectively, for convenience in the presentation. Note that other utterances recorded in the first session were not used for test. Indeed, the utterances recorded in the same session used for training could be employed for test and a very high identification accuracies can be often achieved in this way. However, it does not indicate that the system is robust since there is little variation of speaker's voice recorded in the same session. In contract, the performance of a speaker identification system should be evaluated by utterances recorded in different sessions as suggested by many researchers [17,21,9]. In simulations, an HME classifier worked on a specific feature set to handle the utterances of a digit. As shown in Tables 1–4, simulation results indicate that no specific feature set can outperform other feature sets for all the text (digits) though a comparison is available in terms of the mean identification accuracy. Based on these trained HME classifiers on different feature sets, our linear combination scheme was trained on a cross-validation set consisting of 600 utterances where 30 utterances recorded in the second session were used for each speaker (three utterances of each digit). Other utterances recorded in the second session and all the utterances recorded in the third session were used for test. Accordingly, the testing results are called COMB-1 (test on other 1400 utterances recorded in the second session) and COMB-2 (test on 2000 utterances recorded in the third session), respectively. We show the identification accuracies of our method in Table 5. In contrast to individual HME classifiers on specific feature sets, our method yields considerably better identification accuracies for each digits. In particular, it is evident from simulation results that the identification accuracies were significantly improved when utterances recorded in the third session were used for test. It indicates that our method yields the robust performance.

[2] It refers to that there are two mixture of experts (ME) modules in the structure and eight experts in each ME module [28,30].

Table 1
Text-dependent speaker identification: identification accuracies (%) of the individual HMEs on the delta-cepstrum feature set

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Mean
TEST-1	92.0	88.5	94.5	89.5	96.5	94.5	96.0	85.0	88.5	93.5	91.9
TEST-2	81.5	85.0	90.5	83.5	89.0	80.0	80.5	81.5	79.5	84.5	83.6

Table 2
Text-dependent speaker identification: identification accuracies (%) of the HMEs on the LPC cepstrum feature set

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Mean
TEST-1	89.0	90.0	92.5	90.5	91.5	90.5	91.5	85.5	90.5	95.5	90.7
TEST-2	76.0	85.5	78.5	83.0	88.5	81.0	87.0	78.5	81.5	89.5	82.9

Table 3
Text-dependent speaker identification: identification accuracies (%) of the individual HMEs on the Mel-scale cepstrum feature set

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Mean
TEST-1	87.0	88.0	89.5	87.0	91.5	93.5	94.5	88.5	86.0	91.5	89.7
TEST-2	78.0	88.5	87.5	81.0	86.5	82.0	79.5	76.0	84.5	78.0	82.2

Table 4
Text-dependent speaker identification: identification accuracies (%) of the individual HMEs on the LPC coefficient feature set

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Mean
TEST-1	86.0	90.5	86.5	84.5	89.0	87.5	86.5	81.0	82.5	87.0	86.1
TEST-2	77.0	79.0	80.5	76.5	87.0	81.5	79.0	77.5	75.0	77.5	79.1

Table 5
Text-dependent speaker identification: identification accuracies (%) of our method by combining HMEs on four different feature sets

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Mean
COMB-1	98.5	96.5	95.5	94.5	97.0	97.5	97.5	93.5	95.5	100.0	96.6
COMB-2	95.5	92.5	95.5	96.5	91.5	90.5	97.5	92.5	91.0	97.0	94.0

For comparison, we also adopted two distinct methods to handle the same problem. For comparison with a composite-feature based method commonly used in speaker identification, we conducted an experiment to use a single composite feature for dealing with the problem on the same condition. In the stages of preprocessing and feature, an utterance for a digit was segmented into several frames. For each frame, the composite feature vector was formed by lumping the four corresponding feature vectors together into a 72-dimensional composite feature vector. Then we adopted 10 HMEs with 2–8 structure as classifiers. The results produced by the individual HMEs on the composite feature set are shown in Table 6. Although their performance is slightly better than all the individual HMEs on specific feature sets in general, our method outperforms the composite-feature-based method for all the digit except digit '4' in TEST-2. On the other hand, there exist numerous methods of combining multiple classifiers on different feature sets. For comparison with the existing combination methods, we also conducted an experiment by using an existing method called Bayesian reasoning combination to deal with the same problem. The Bayesian reasoning combination scheme proposed by Xu et al. [46] has been used for combining multiple classifiers on different feature sets. Previous comparative studies showed that the method yielded the best recognition results for several benchmark OCR problems among numerous combination methods ranging from voting to evidence-reasoning based combination methods [46,35]. In simulations, we also used the same cross-validation set and test sets used in our method for training and test. For training, the confusion matrix containing the prior knowledge on each classifier was calculated based on outputs of those HME classifiers in terms of the cross-validation set [46]. Using the confusion matrix, the combination scheme makes an MAP decision on the basis of all the HME classifiers' outputs. In the Bayesian reasoning combination method, a threshold is required to reject an unknown sample in some cases. In our simulations, the rejection threshold was set as zero for convenience in comparison. In addition, the results produced by the Bayesian reasoning combination method are also called COMB-1 and COMB-2, respectively, corresponding to two aforementioned testing sets. Accordingly, these results are shown in Table 7. In comparison with the Bayesian reasoning combination method, our method yields better performance for all the digits except for '1' and '8' where the Bayesian reasoning combination method is slightly better than our method in COMB-2 though our method is better for the same digits in COMB-1.

In practice, the above testing method is not directly used in a speaker identification system because the utterance of a single digit is too short to produce a robust result.

Table 6
Text-dependent speaker identification: identification accuracies (%) of the individual HMEs on the composite feature set consisting of four different feature sets

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Mean
TEST-1	93.0	92.5	87.0	88.5	96.0	97.0	93.5	91.5	91.5	98.5	92.9
TEST-2	88.0	90.5	91.0	87.5	94.5	87.0	90.5	84.0	87.5	94.5	89.5

Table 7
Text-dependent speaker identification: identification accuracies (%) of the Bayesian reasoning combination method by combining HMEs on four different feature sets

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Mean
COMB-1	98.0	96.5	94.5	92.5	96.0	97.0	97.5	91.5	93.5	100.0	95.7
COMB-2	94.5	93.5	94.5	95.5	90.5	89.0	95.5	91.5	91.5	96.5	93.3

Table 8
Text-dependent speaker identification: experimental results produced by the our method in terms of the sequence-based test

Test no.	500	1000	2000	3000	4000	5000
Recognition No.	500	1000	2000	2999	3998	4997
Substitution No.	0	0	0	0	1	1
Rejection No.	0	0	0	1	1	2

Table 9
Text-dependent speaker identification: experimental results produced by the composite-based method in terms of the sequence-based test

Test no.	500	1000	2000	3000	4000	5000
Recognition No.	500	999	1998	2994	3993	4989
Substitution No.	0	0	1	3	3	3
Rejection No.	0	1	1	3	4	8

Table 10
Text-dependent speaker identification: experimental results produced by the Bayesian reasoning combination method in terms of the sequence-based test

Test no.	500	1000	2000	3000	4000	5000
Recognition No.	500	1000	2000	2999	3997	4996
Substitution No.	0	0	0	0	1	1
Rejection No.	0	0	0	1	2	3

A so-called sequence-based test method is often adopted in a practical speaker identification system based on the single-digit based test to identify personal identity [13]. In our simulations, we first produced a sequence consisting of five digits at random (it may be viewed as a password), then asked a speaker to utter the

digit-sequence on line. For each digit in the sequence, obviously, an identification result was available based on the single-digit based method. After obtaining all the five results, the system polled a vote with the principle of majority that an unknown speaker can be identified only when there are at least three same identification results for the speaker; otherwise, the system rejects the unknown utterance. Due to the limited space, here we only report the experimental results on the composite-feature-based method and two combination methods in terms of the sequence-based test. As shown in Tables 8–10, the results of the sequence-based test also indicate that our method outperforms other two methods for text-dependent speaker identification.

In general, simulation results show that our method outperforms not only individual HME classifiers on specific feature sets and the composite-feature based method but also the Bayesian reasoning combination method in text-dependent speaker identification.

4.2. Results on text-independent speaker identification

Text-independent speaker identification is a more difficult problem. Because the text may be arbitrary, all the template matching techniques are not applicable. As a result, speaker's features play an especially important role in text-independent speaker identification. In order to further evaluate its performance, we have also applied our method to a text-independent speaker identification problem.

In simulations, the database was a subset of the standard speech database in China. This set represents 20 speakers (10 male and 10 female) of the same (Mandarin) dialect. The utterances in the database were recorded in three separate sessions. In the first session, 10 different phonetically rich sentences were uttered by each speaker. The average length of the sentences was about 4.5 s. In the second and the third sessions, five different sentences were uttered by each speaker, respectively. Their average lengths of the sentences were about 4.4 and 5.0 s, respectively. All utterances were recorded in a quiet room and sampled at 11.025 KHz sampling frequency in 16-bit precision. In simulations, we adopted three speech spectral feature sets commonly used in text-independent speaker identification [17,21,9], i.e., 24-order LPC based cepstrum (24-LPCCEP), 20-order Mel-scale cepstrum (24-MELCEP), and 24-order LPC coefficients (24-LPCCOE). For text-independent speaker identification, it is generally agreed that the voiced parts of an utterance, especially vowels and nasals, are more effective in contrast to the unvoiced parts [2,39,37,9]. In simulations, therefore, only the voiced parts of a sentence remained regardless of their contents by using a common energy measure [36]. The length of the Hamming analysis window was 64 ms without overlapping. Here, we emphasize that the size of analysis window was slightly larger than the common one (normally $16 \sim 32$ ms) since it was found that the identification performance was degraded with a normal analysis window [24]. Whenever the short-time energy of a speech frame was higher than a pre-specified threshold, spectral feature vectors would be calculated based on the different feature extraction methods. In addition, samples were also pre-emphasized by the filter $H(z) = 1 - 0.97z^{-1}$. The three different feature sets were achieved after the

aforementioned processing. For utterances of all 20 speakers, total numbers of speech frames were 10057 frames, 4270 frames, and 4604 frames corresponding to utterances recorded in three different sessions, respectively. To simplify the presentation, we denote SET-i as the data set recorded in session i ($i = 1,2,3$).

The evaluation of a speaker identification experiment was conducted in the following manner as suggested by Reynolds [37]. After feature extraction, the testing speech is to produce a sequence of feature vectors denoted as $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_T\}$. The sequence of feature vectors is divided into overlapping segments of S feature vectors. The first two segments from a sequence would be

$$\overbrace{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_S}^{\text{Segment 1}} \boldsymbol{f}_{S+1}, \boldsymbol{f}_{S+2}, \cdots, \boldsymbol{f}_T$$

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

Thus, a test segment length of 6.4 s should correspond to $S = 100$ feature vectors for a 6.4 ms frame rate in accordance with the above definition. First of all, we chose $S = 100$ as the test segment length; accordingly, total numbers of segments were 2290 in SET-2 and 2624 in SET-3, respectively, for utterances of all 20 speakers. Each segment of S vectors was treated as a separate testing utterance and identified using the classification procedure of a classifier. Using a segment, the system produced either an identification result or a rejection. Based on the segment testing method, an unknown speaker was identified only when there were at least 50% input vectors in the segment that produced the same identification results for the speaker; otherwise, the system rejected the unknown speaker. The above steps were repeated for testing utterances from each speaker in the population. The final performance evaluation was then computed according to *identification rate*, *substitution rate* and *rejection rate* defined, respectively, as

$$\text{identification rate} = \frac{\sharp \text{ correctly identified segments}}{\text{total } \sharp \text{ of segments}} \times 100\%, \tag{18a}$$

$$\text{substitution rate} = \frac{\sharp \text{ incorrectly identified segments}}{\text{total } \sharp \text{ of segments}} \times 100\%, \tag{18b}$$

$$\text{rejection rate} = 100\% - \text{identification rate} - \text{substitution rate}. \tag{18c}$$

In the sequel, Identification, Substitution, and Rejection are the abbreviations for identification rate, substitution rate and rejection rate for simplicity.

In simulations, we still adopted the HME architecture as individual classifiers because of its good performance in classification [30]. For model selection in the current problem, we examined six different HME structures ranging for two-level to

four level using a two-fold cross-validation method. Finally, we chose HME with 2–9 structure for the problem. We first trained three individual HMEs with 2–9 structure on the three different feature sets, respectively, using the EM algorithm [30,15]. All the short-time speech frames in SET-1 were used to train individual HME classifiers. Moreover, all the short-time speech frames in SET-2 as a cross-validation data set were used to train our linear combination scheme. For test, all the short-time speech frames in SET-3 were used. As shown in Table 11, simulation results indicate that the HME on the Mel-scale cepstrum feature set slightly outperforms HMEs on the other two feature sets. In contrast, our method yields significantly improved performance. The identification rate of our method is at least 6.7% higher than any individual HME on a specific feature set, while the substitution rate of our method is zero in terms of 2.2% rejection rate.

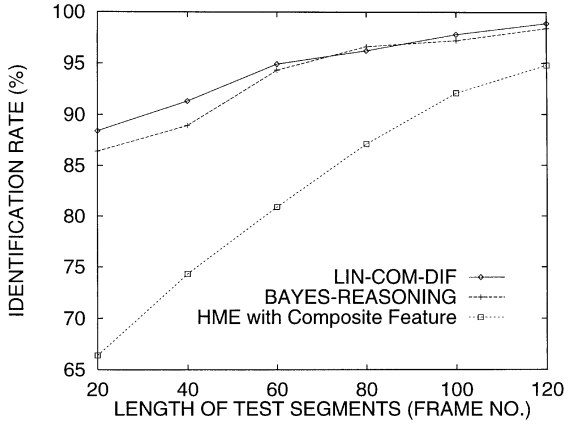
Similarly, we also conducted simulations for comparison with the composite-feature based method and the Bayesian reasoning combination method. In the composite-feature based method, an HME with 2–9 structure was also applied to the same task based on a single composite feature set formed from the three different feature sets. The testing result on SET-3 is also shown in Table 11. From the simulation results, it is evident that our method is superior to the composite-based method. Furthermore, we applied the Bayesian reasoning combination method to the same task. The prior knowledge or confusion matrix was also achieved on the basis of speech frames in SET-2 and the testing result on SET-3 is shown in the same table. Simulation results also indicate that our method is slightly better than the Bayesian reasoning combination method. In particular, both combination methods produce zero substitution rate with a small rejection rates.

It is well known that the performance of a speaker identification system depends upon the length of text used. That is, a short text used for test often causes the performance of a speaker identification system to be degraded, while a long text may often lead to a better performance. In practice, however, a long text can affect identification speed and result in a unfriendly user interface. Therefore, the robustness

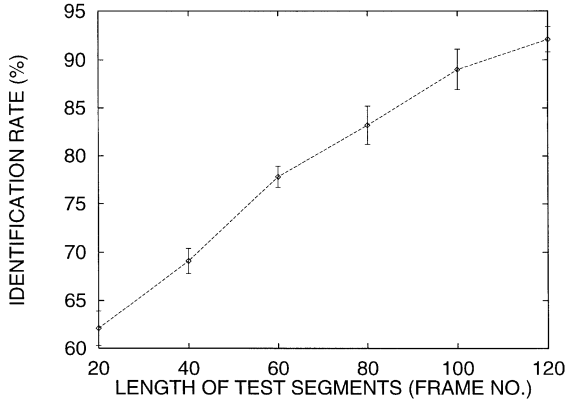
Table 11
Text-independent speaker identification: experimental results (%) produced by individual HMEs on specific feature sets, our method (LIN-COM-DIF), the individual HME on the composite feature set, and the Bayesian reasoning combination method (BAYES-REASONING). Each test segment contains 100 short-term speech frames in this test

Classifier (Features)	Identification	Substitution	Rejection
HME (24-LPCCEP)	89.5	5.2	5.3
HME (20-MELCEP)	91.1	3.1	5.8
HME (24-LPCCOE)	86.9	5.7	7.4
LIN-COM-DIF	97.8	0.0	2.2
HME (composite feature)	92.1	3.3	4.6
BAYES-REASONING	97.2	0.0	2.8

is another critical aspect to evaluate the performance of a speaker identification system. For this purpose, we conducted some simulations using different testing segment lengths generated from SET-3. As a result, identification and substitution rates produced by different methods are illustrated in Figs. 3 and 4, respectively. Based on the results, basically, our method outperforms other methods used in our simulations. It is evident that our method yields higher identification rates and lower substitution rates when short speech testing segments are used. We emphasize that our method is not only significantly better than an individual HME on either a



(a)



(b)

Fig. 3. Text-independent speaker identification: identification rates produced by different methods on speech test segments with different lengths. (a) Results produced by the composite-feature-based method, the Bayesian reasoning combination method (BAYES-REASONING), and our method (LIN-COM-DIF). (b) Results produced by individual HME classifiers on specific feature sets.

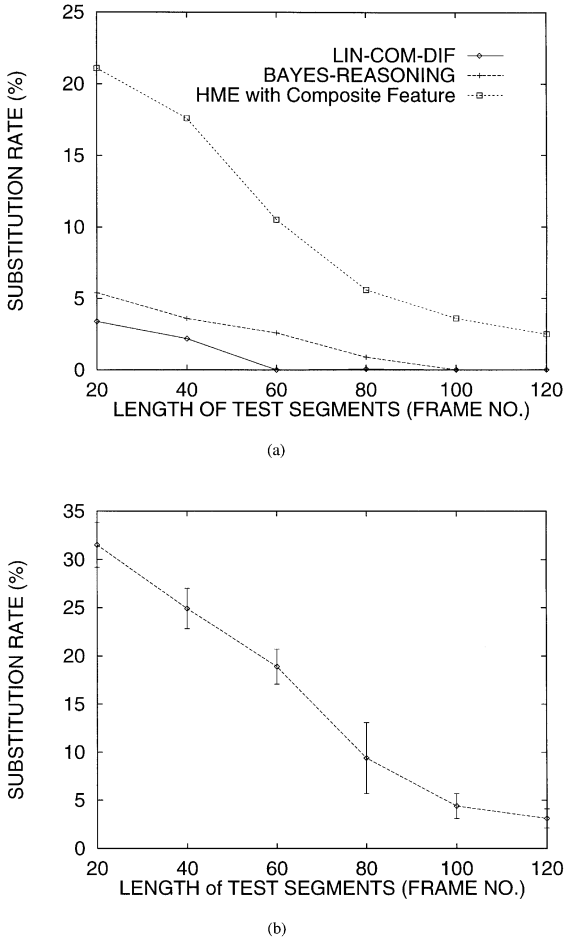


Fig. 4. Text-independent speaker identification: substitution rates produced by different methods on speech test segments with different lengths. (a) Results produced by the composite-feature-based method, the Bayesian reasoning combination method (BAYES-REASONING), and our method (LIN-COM-DIF). (b) Results produced by individual HME classifiers on specific feature sets.

specific feature set or the composite feature set but also outperforms the Bayesian reasoning combination method in robustness. In contrast to the Bayesian reasoning combination method, in particular, the identification rates of our methods were at least 2.0% higher, while its substitution rates were at least 1.5% lower when the speech testing segments containing 20 and 40 frames were used, respectively.

In general, comparative results indicate that our method also yields the best performance in identification rate and robustness for text-independent speaker identification.

5. Concluding remarks

We have presented a novel method of combining multiple probabilistic classifiers on different feature sets under the framework of linear opinion pools. In the proposed method, soft competition is adopted as an automatic feature rank mechanism for use of different feature sets in an optimal way. Based on the soft competition mechanism, a generalized finite mixture model is proposed as a linear combination scheme and an EM learning algorithm is also developed for parameter estimation in the linear combination scheme. To demonstrate its effectiveness, we apply the proposed method to a typical real world task, viz., speaker identification, which often needs to simultaneously use different feature sets for robustness. Simulation results show that our linear combination scheme outperforms individual HME classifiers on specific feature sets, the composite-feature based method, and a sophisticated combination method called Bayesian reasoning combination in speaker identification.

Several issues with respect to our linear combination scheme are worth to be addressed. First of all, our method can be viewed as an extension of the existing models on linear opinion pools with weights as veridical probabilities [28,44] in terms of pattern classification on different feature sets. In those models, the same feature vectors are input to both the combination scheme and classifiers, and, therefore, a composite-feature based method is inevitably used for a task of pattern classification on different feature sets. Moreover, those models can be also regarded as a special case of our linear combination scheme. When a robust feature set can be available as illustrated in Fig. 1a, the same feature vector is input to different classifiers and our linear combination scheme where there is only a subscheme. In this case, our linear combination scheme is an equivalent to the model proposed by Xu and Jordan [44]. Next, our method is different from a recent combination method proposed by Tresp and Taniguchi [41]. Their method is also a novel scheme of combining estimators using non-constant weighting functions under the framework of linear opinion pools. In their linear combination scheme, the weighting functions are also dependent on the input. However, estimators in their method are allowed to work on disjoint regions of input space and the method is applied in a regression task for better generalization [41]. In contract, our linear combination scheme works on different representation forms of input space and all the estimators in our method still work on the same input space though different feature sets are used for pattern classification. This feature significantly distinguishes our method from their method in terms of motivation and application. It should be pointed out that our method may be used in several circumstances. In this paper, we only use the same type of classifiers, i.e., HME architecture, and only one classifier to deal with a specific feature set for pattern classification on different feature sets. In fact, our method can be used to combine multiple different types of classifiers. As a result, an extensive study on this topic has been done and experimental results show that our method also yields good performance in different circumstances [11]. On the other hand, Kanal [31] argued that the research on combination of multiple classifiers provides a new perspective of pattern recognition. His argument suggested that we can build a number of different and complementary classifiers instead of developing a single high-performance classifier.

Each classifier itself may not produce the desired performance, but an optimal combination of such classifiers may lead to a highly reliable performance. More recently, this issue has been examined by different researchers [19,8,29] and their methods have turned out to be a hopeful way to develop new pattern recognition systems. In our ongoing research, we shall investigate the performance of our method in combining "weak" probabilistic classifiers for pattern classification on different feature sets. In addition, regularization techniques will be introduced to our linear combination scheme to improve its generalization capability. We expect that our method will be successfully applied in more problems of pattern classification on different feature sets.

Acknowledgements

Authors would like to thank L. Xu for a valuable discussion, L. Wang for her help in simulations, and two anonymous referees for their constructive comments that significantly improved the presentation of this paper. This work was partially supported by National Science Foundation in China.

Appendix A.

In this appendix, we present the detailed derivation of the EM algorithm described in Section 3 for parameter estimation in our linear combination scheme.

In order to derive the EM algorithm, we assume that N classifiers have been already trained on K ($K \leqslant N$) different feature sets extracted from a training data set for a given task. For a cross-validation set $\mathscr{X} = \{(D^{(t)}, \boldsymbol{y}^{(t)})\}_{t=1}^{\mathrm{T}}$, K different feature sets, $\{\boldsymbol{x}_1^{(t)}\}_{t=1}^{\mathrm{T}}, \ldots, \{\boldsymbol{x}_K^{(t)}\}_{t=1}^{\mathrm{T}}$, can be extracted from the input data set, $\{D^{(t)}\}_{t=1}^{\mathrm{T}}$, with the same feature extraction methods. For the generalized finite mixture model in Eq. (5), we apply the EM algorithm to estimate all parameters in $\Phi = \{(\alpha_{kj}(\boldsymbol{x}_k), \beta_k) | j = 1, \ldots, N; k = 1, \ldots, K\}$. For use of the EM algorithm, we introduce a set of indicators $\mathscr{I} = \{(II_j^{(t)}, I_k^{(t)}) | j = 1, \ldots, N; k = 1, \ldots, K; t = 1, \ldots, T\}$ as missing data or unobserved data to the observed data $\mathscr{X}$. These indicators are defined as

$$II_j^{(t)} = \begin{cases} 1 & \text{if the final decision is made by classifier } e_j, \\ 0 & \text{otherwise.} \end{cases} \tag{A.1a}$$

$$I_k^{(t)} = \begin{cases} 1 & \text{if weights for combination are generated by the } i\text{th subscheme,} \\ 0 & \text{otherwise.} \end{cases} \tag{A.1b}$$

Moreover, these indicators satisfy the following conditions: $\sum_{j=1}^{N} II_j^{(t)} = 1$ and $\sum_{k=1}^{K} I_k^{(t)} = 1$, for $t = 1, \ldots, T$. Thus, the complete data set is achieved and composed of both the observed data set $\mathscr{X}$ and the missing data set $\mathscr{I}$. In general, an EM algorithm

consists of two consecutive steps: Expectation step (E-step) and Maximization step (M-step) [16].

In the E-step, the task is to acquire the expectation of the missing data on the condition of the observed data. At the sth iteration, the expectation is calculated based on the Bayesian rule as follows.

$$
\begin{aligned}
E[II_j^{(t)}, I_k^{(t)} | \mathscr{X}] &= P(II_j^{(t)} = 1, I_k^{(t)} = 1 | \boldsymbol{y}^{(t)}, D^{(t)}, \Phi^{(s)}) \\
&= \frac{P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(II_j^{(t)} = 1, I_k^{(t)} = 1 | D^{(t)}, \Phi^{(s)})}{P(\boldsymbol{y}^{(t)} | D^{(t)}, \Phi^{(s)})} \\
&= \frac{P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(II_j^{(t)} = 1 | I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(I_k^{(t)} = 1 | D^{(t)}, \Phi^{(s)})}{P(\boldsymbol{y}^{(t)} \ | D^{(t)}, \Phi^{(s)})} \\
&= \frac{P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(II_j^{(t)} = 1 | I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(I_k^{(t)} = 1 | D^{(t)}, \Phi^{(s)})}{P(\boldsymbol{y}^{(t)} | D^{(t)}, \Phi^{(s)})} \\
&= \frac{\alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) \beta_k^{(s)} P(\boldsymbol{y}^{(t)} | \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)}{\sum_{j=1}^{N} \sum_{k=1}^{K} \beta_k^{(s)} \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) P(\boldsymbol{y}^{(t)} | \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)}
\end{aligned}
\tag{A.2}
$$

and

$$
\begin{aligned}
E[I_k^{(t)} | \mathscr{X}] &= P(I_k^{(t)} = 1 \,|\, \boldsymbol{y}^{(t)}, D^{(t)}, \Phi^{(s)}) \\
&= \frac{P(\boldsymbol{y}^{(t)} \,|\, I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(I_k^{(t)} = 1 \,|\, D^{(t)}, \Phi^{(s)})}{P(\boldsymbol{y}^{(t)}, D^{(t)} \,|\, \Phi^{(s)})} \\
&= \frac{\sum_{j=1}^{N} P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(II_j^{(t)} = 1 \,|\, I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(I_k^{(t)} = 1 | D^{(t)}, \Phi^{(s)})}{P(\boldsymbol{y}^{(t)} | D^{(t)}, \Phi^{(s)})} \\
&= \frac{\sum_{j=1}^{N} P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(II_j^{(t)} = 1 | I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) P(I_k^{(t)} = 1 | D^{(t)}, \Phi^{(s)})}{P(\boldsymbol{y}^{(t)} | D^{(t)}, \Phi^{(s)})} \\
&= \frac{\sum_{j=1}^{N} \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) \beta_k^{(s)} P(\boldsymbol{y}^{(t)} | \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)}{\sum_{j=1}^{N} \sum_{k=1}^{K} \beta_k^{(s)} \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}) P(\boldsymbol{y}^{(t)} | \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)}.
\end{aligned}
\tag{A.3}
$$

Note that the statistical model of classifier e_j, $P(\boldsymbol{y} | \boldsymbol{x}_{k_j}^{(t)}, \Theta_j)$, is independent of indicator $I_k^{(t)}$ in terms of the definition in Eq. (A.1a). Therefore, we may apply $P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, I_k^{(t)} = 1, D^{(t)}, \Phi) = P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, D^{(t)}, \Phi)$ in the above derivation. In addition, the following relations have been used in the last step of Eqs. (A.2) and (A.3) in accordance with the definition of indicators:

$$
\begin{aligned}
&P(II_j^{(t)} = 1 | I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = \alpha_{kj}^{(s)}(\boldsymbol{x}_k^{(t)}), \\
&P(I_k^{(t)} = 1 | D^{(t)}, \Phi^{(s)}) = \beta_k^{(s)}, \\
&P(\boldsymbol{y}^{(t)} | II_j^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = P(\boldsymbol{y}^{(t)} | \boldsymbol{x}_{k_j}^{(t)}, \Theta_j).
\end{aligned}
$$

Therefore, the posterior probabilities, $h_k^{(s)}(\boldsymbol{y}^{(t)}|\boldsymbol{x}_k^{(t)})$ and $h_{kj}^{(s)}(\boldsymbol{y}^{(t)}|\boldsymbol{x}_k^{(t)})$, used in the EM algorithm are achieved as $h_k^{(s)}(\boldsymbol{y}^{(t)}|\boldsymbol{x}_k^{(t)}) = E[I_k^{(t)}|\mathscr{X}]$ and $h_{kj}^{(s)}(\boldsymbol{y}^{(t)}|\boldsymbol{x}_k^{(t)}) = E[I_k^{(t)}, II_j^{(t)}|\mathscr{X}]$.

In the M-step, the task is to maximize a set of simplified objective functions derived from the log-likelihood function based on the posterior probabilities achieved in the E-step. In our problem, we simplify the log-likelihood function in Eq.(Eq. (10)) using a trick suggested by [45]. As a result, we rewrite the generalized finite mixture model into the following equivalent form:

$$P(\boldsymbol{y}, D) = P(\boldsymbol{y}|D,\Phi)P(\boldsymbol{x}_k,\Psi) = \sum_{j=1}^{N}\sum_{k=1}^{K}\beta_k\lambda_{kj}P(\boldsymbol{x}_k,\Psi_{kj})P(\boldsymbol{y}|\boldsymbol{x}_{k_j},\Theta_j), \tag{A.4}$$

where $P(\boldsymbol{x}_k,\Psi) = \sum_{j=1}^{N}\lambda_{kj}P(\boldsymbol{x}_k,\Psi_{kj})$ and $P(\boldsymbol{x}_k,\Psi_{kj}) = \Omega(\boldsymbol{x}_k,\Psi_{kj})$. $\Omega(\boldsymbol{x}_k,\Psi_{kj})$ is as same as defined in Eq. (7). Utilizing Eq. (A.4), we can simplify the log-likelihood function in Eq. (10) as the following objective functions:

$$Q(\Psi_{kj}) = \sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{k=1}^{K} h_{kj}^{(s)}(\boldsymbol{y}^{(t)}|\boldsymbol{x}_k^{(t)})\log\Omega(\boldsymbol{x}_k,\Psi_{kj}), \tag{A.5}$$

$$Q(\lambda_{kj}) = \sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{k=1}^{K} h_{kj}^{(s)}(\boldsymbol{y}^{(t)}|\boldsymbol{x}_k^{(t)})\log\lambda_{kj}, \tag{A.6}$$

and

$$Q(\beta_k) = \sum_{t=1}^{T}\sum_{k=1}^{K} h_k^{(s)}(\boldsymbol{y}^{(t)}|\boldsymbol{x}_k^{(t)})\log\beta_k. \tag{A.7}$$

For our problem, the remaining task in the M-step is to find a new estimate of each parameter by maximizing the above separate objective functions as

$$\Psi_{kj}^{(s+1)} = \arg\max_{\Psi_{kj}} Q(\Psi_{kj}), \tag{A.8}$$

$$\lambda_{kj}^{(s+1)} = \arg\max_{\lambda_{kj}} Q(\lambda_{kj}) \quad \text{s.t. } \textstyle\sum_{j=1}^{N}\lambda_{kj} = 1, \tag{A.9}$$

and

$$\beta_k^{(s+1)} = \arg\max_{\beta_k} Q(\beta_k) \quad \text{s.t. } \textstyle\sum_{k=1}^{K}\beta_k = 1. \tag{A.10}$$

Thanks to the joint distribution from in Eq. (A.4) and the basis function modeled as a Gaussian distribution in Eq. (7), all the optimization problems in Eqs. (A.8), (A.9) and (A.10), become analytically solvable and their solutions have been presented in Eqs. (13)–(16).

References

[1] C. Agnew, Multiple probability assessments by dependent experts, J. Amer. Statist. Assoc. 80 (3) (1985) 343–347.

[2] B.S. Atal, Effectiveness of linear prediction characteristics of the speech waves for automatic speaker identification and verification, J. Acoust. Soc. Amer. 55 (6) (1974) 1304–1312.

[3] R. Battiti, A.M. Colla, Democracy in neural nets: voting schemes for classification, Neural Networks 7 (4) (1994) 691–708.
[4] Y. Bennani, A modular and hybrid connectionist system for speaker identification, Neural Comput. 7 (4) (1995) 791–798.
[5] Y. Bennani, P. Gallinari, Connectionist approaches for automatic speaker recognition, Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, 1994, pp. 95–102.
[6] A. Blum, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (2) (1997) 245–272.
[7] L. Breiman, Stacked regression, Tech. Rep. TR-367, Department of Statistics, University of California, Berkeley, 1992.
[8] L. Breiman, Bagging predictors, Mach. Learning 26 (2) (1996) 123–140.
[9] F.P. Campbell, Speaker recognition: a tutorial, Proc. IEEE 85 (9) (1997) 1437–1463.
[10] S. Chatterjee, S. Chatterjee, On combining expert opinions, Amer. J. Math. Management Sci. 7 (1) (1987) 271–295.
[11] K. Chen, L. Wang, H. Chi, Methods of combining multiple classifiers with different features and their applications to text-indpendent speaker identification, Int. J. Pattern Recognition Artif. Intell. 11 (3) (1997) 417–445.
[12] K. Chen, D. Xie, H. Chi, Speaker identification based on hierarchical mixture of experts, Proc. World Congress on Neural Networks, Washington DC, 1995, pp. I493–I496.
[13] K. Chen, D. Xie, H. Chi, A modified HME architecture for text-dependent speaker identification, IEEE Trans. Neural Networks 7 (5) (1996) 1309–1313.
[14] K. Chen, D. Xie, H. Chi, Speaker Identification using time-delay HMEs, Int. J. Neural Systems 7 (1) (1996) 29–43.
[15] K. Chen, L. Xu, H. Chi, Improved learning algorithms for mixtures of experts in multiway classification, Neural Networks (1996) (in revision).
[16] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum-likelihood from incomplete data via the EM algorithm, J. Roy. Statist. Soc. B 39 (1) (1977) 1–38.
[17] G. Doddington, Speaker recognition – identifying people by their voice, Proc. IEEE 73 (11) (1986) 1651–1664.
[18] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[19] Y. Freund, R. Schapire, Experiments with a new boosting algorithm, Proc. 13th Int. Conf. Machine Learning, 1996, pp. 148–156.
[20] S. Furui, An overview of speaker recognition technology, Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, 1994, pp. 1–9.
[21] S. Furui, Recent advances in speaker recognition, Pattern Recognition Lett. 18 (9) (1997) 859–872.
[22] A. Gelfand, B. Ballick, D. Dey, Modeling expert opinion arising as a partial probabilistic specification, J. Amer. Statist. Assoc. 90 (5) (1995) 598–604.
[23] S. Hashem, Optimal linear combinations of neural networks, Tech. Rep. SMS 94-4, School of Industrial Engineering, Purdue University, 1993.
[24] J. He, L. Liu, G. Palm, A text-independent speaker identification system based on neural networks, Proc. Int. Conf. Spoken Language Processing, Yokohama, 1994, pp. 181–185.
[25] T. Ho, J. Hull, S. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.
[26] Y. Huang, C. Suen, A method of combining multiple experts for the recognition of unconstrained handwritten numerals, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1) (1995) 90–94.
[27] R.A. Jacobs, Methods for combining experts' probability assessments, Neural Comput. 7 (5) (1995) 867–888.
[28] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Comput. 3 (1) (1991) 79–87.
[29] C. Ji, S. Ma, Combinations of weak classifiers, IEEE Trans. Neural Networks 7 (1) (1997) 32–42.
[30] M.I. Jordan, R.A. Jocobs, Hierarchical mixture of experts and EM algorithm, Neural Comput. 6 (2) (1994) 181–214.

[31] L. Kanal, On pattern, categories, and alternative realities, Pattern Recognition Lett. 14 (3) (1993) 241–255.
[32] M. LeBlanc, R. Tibshirani, Combining estimates in regression and classification, Tech. Rep., Department of Preventive Medicine and Biostatistics, University of Toronto, 1993.
[33] R. Meir, Bias, variance, and the combination of estimators, Tech. Rep. 922, Department of Electrical Engineering, Technion, Haifa, Israel, 1994.
[34] D. O'Shaughnessy, Speaker recognition, IEEE ASSP Mag. 3 (4) (1986) 4–17.
[35] 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.
[36] L.R. Rabiner, B.H. Juang, Fundamentals of Speech Recognition, Englewood Cliffs, Prentice-Hall, New Jersey, 1993.
[37] 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.
[38] G. Rogova, Combining the results of several neural network classifiers, Neural Networks 7 (5) (1994) 777–781.
[39] M.R. Sambur, Selection of acoustic features for speaker identification, IEEE Trans. Acoust. Speech Signal Process. 23 (1) (1975) 176–182.
[40] F.K. Soong, A.E. Rosenberg, On the use of instantaneous and transitional spectral information in speaker recognition, IEEE Trans. Acoust. Speech Singal Process. 36 (6) (1988) 871–879.
[41] V. Tresp, M. Taniguchi, Combining estimators using non-constant weithting functions, in: G. Tesauro, D. Tourezsky, T. Leen (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 1995, pp. 419–426.
[42] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang, Phoneme recognition using time-delay neural networks, IEEE Trans. Acoust. Speech Signal Process. 37 (3) (1989) 328–339.
[43] D.H. Wolpert, Stacked generalization, Neural Networks 5 (2) (1992) 241–259.
[44] L. Xu, M.I. Jordan, EM learning on a generalized finite mixture model for combining multiple classifiers, Proc. World Congress on Neural Networks, San Diego, 1993, pp. IV227–IV230.
[45] L. Xu, M.I. Jordan, G.E. Hinton, An alternative model for mixture of experts, in: G. Tesauro, D. Tourezsky, T. Leen (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 1995.
[46] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. System Man. Cybernet. 23 (3) (1992) 418–435.

Ke Chen received his B.S. and M.S. in computer science from Nanjing University in 1984 and 1987, respectively, as well as his Ph.D. in computer science and engineering from Harbin Institute of Technology in 1990. From 1990 to 1992, he was a postdoctoral researcher at Tsinghua University. During 1992-1993 he was a postdoctoral fellow of *Japan Society for Promotion of Sciences* and worked at Kyushu Institute of Technology. From 1997 to 1998, he was a research scientist at The Ohio State University. He joined Peking University in 1993, and is now a full professor of Information Science. He has published over 50 technical papers in refereed journals and international conferences. His current research interest includes neural computation and its applications in machine perception. Dr. Chen is a member of IEEE and a senior member of CIE.

Huisheng Chi graduated from department of radio and electronics at Peking University in 1964 (six-year system) and has been working in the university since then. Major research interests are in satellite communications, digital communications and speech signal processing. In recent years, the research projects conducted by him involved in the neural network auditory model and speaker identification systems. He received *Neural Network Leadership Award* in 1994 and 1995 from INNS. he is a deputy president of Peking University, where he is also a full professor of information science. He serves as an associate editor of *IEEE Transactions on neural networks*. Prof. Chi is a senior member of IEEE, a member of appraisal group of NSFC and SEC, fellow of CIE and CIC, and a vice chairman of CNNC and CAGIS.