ELSEVIER

Pattern Recognition Letters 19 (1998) 545–558

Pattern Recognition Letters

A connectionist method for pattern classification with diverse features

Ke Chen [a,b,*]

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

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

Received 18 August 1997; revised 15 April 1998

Abstract

A novel connectionist method is proposed to simultaneously use diverse features in an optimal way for pattern classification. Unlike methods of combining multiple classifiers, a modular neural network architecture is proposed through use of soft competition among diverse features. Parameter estimation in the proposed architecture is treated as a maximum likelihood problem, and an Expectation-Maximization (EM) learning algorithm is developed for adjusting the parameters of the architecture. Comparative simulation results are presented for the real world problem of speaker identification.

Keywords: Classification with diverse features; Mixture of experts; Expectation-Maximization (EM) algorithm; Soft competition; Speaker identification

1. Introduction

A general pattern classification process is usually composed of three stages, i.e., preprocessing, feature extraction, and classification. In the stage of feature extraction, in particular, there may be numerous different methods so that several diverse features can be extracted from the same raw data. To a large extent, each feature can independently represent the original data, but none of them is totally perfect for practical applications. Moreover, there seems to be no simple way to measure which kind of feature is optimal for a pattern classification task. For this kind of pattern classification tasks, diverse features often need to be jointly used in order to achieve robust performance. In this paper, we call this kind of pattern classification tasks it *classification with diverse features.*

To our knowledge, so far there exist two distinct methods for classification with diverse features. One is the use of a composite feature formed by lumping diverse features together somehow, and the other is combination

* E-mail: kchen@cis.ohio-state.edu.

0167-8655/98/$19.00
PII: S0167-8655(98)00055-5

of multiple classifiers that have been already trained on diverse feature sets respectively. For the use of a composite feature, however, there are at least three problems as follows:

- curse of dimensionality; its dimensionality is higher than that of any component feature.
- difficulty in formation; it is difficult to lump several features together due to their diversified forms.
- redundancy; those component features are usually not independent.

In general, therefore, the use of a composite feature does not lead to a significantly improved performance. On the other hand, combination of multiple classifiers has been studied in the pattern recognition community and yields improved performance (Xu et al., 1992; Suen et al., 1993; Ho et al., 1994; Huang and Suen, 1995). Furthermore, more recent studies show that better performance can be achieved by combining multiple classifiers with diverse features (Xu et al., 1992; Perrone, 1993; Chen et al., 1997) in contrast to the combination of multiple classifiers with the same feature. For most of the combination methods, however, a large amount of data is usually required to train both individual classifiers and a combination scheme. In addition, the combination methods are also viewed as a kind of sub-optimal sequential learning procedure (Fogelman-Soulie et al., 1993).

In this paper, we propose a novel method for classification with diverse features. The idea underlying the proposed method is to use diverse features for classification in the manner of soft competition. In contrast to the winner-take-all mechanism, soft competition is a concept that a competitor and its rivals can work for a specific task together, but the winner plays a more important role than the losers. Recently, Jacobs et al. (1991) proposed a modular neural network architecture called *mixture of experts* (ME) for supervised learning. The ME architecture is based on the divide-and-conquer principle, in which a large, hard to solve problem is broken up into many smaller, easier to solve problems. The use of the divide-and-conquer principle makes the ME architecture yield good performance and allows fast training. Although the ME architecture has been successfully applied to several supervised learning tasks (Jordan and Jacobs, 1994; Chen et al., 1996a,b), it can only use a composite feature for classification with diverse features, since both gating and expert networks need to receive the same input. In this paper, we propose an *alternative mixture of experts* (AME) architecture by introducing a soft competition mechanism for the effective use of diverse features. In the AME architecture, parameter estimation is treated as a maximum likelihood problem and an *Expectation-Maximization* (EM) algorithm is developed to adjust the parameters. In terms of maximum likelihood learning, the AME architecture simultaneously uses diverse features in an optimal way. Therefore, it provides a novel connectionist method for classification with diverse features. To evaluate the performance, we have applied the proposed method to the real world problem of speaker identification, and simulation results show the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 describes the AME architecture and the EM algorithm. Section 3 reports simulation results and conclusions are drawn in Section 4.

2. Architecture and learning algorithm

2.1. Motivation

For an input sample $D^{(t)}$ in the data set $\mathscr{X} = \{D^{(t)}, \boldsymbol{y}^{(t)}\}_{t=1}^{T}$, we assume that K feature vectors, $\boldsymbol{x}_1^{(t)}, \ldots, \boldsymbol{x}_K^{(t)}$, can be achieved from the sample $D^{(t)}$ using K ($K > 1$) different feature extraction methods. A question can be raised: which one is the optimal feature of the sample $D^{(t)}$ among its K feature vectors? Prior to addressing an answer to the question, we first introduce a set of binary indicator variables to represent the optimal feature. An indicator $I_k^{(t)}$ for $\boldsymbol{x}_k^{(t)}$ is defined as

$$I_k^{(t)} = \begin{cases} 1 & \text{if } \boldsymbol{x}_k^{(t)} \text{ is the optimal feature of } D^{(t)}, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $\sum_{k=1}^{K} I_k^{(t)} = 1$. If we always use the optimal feature to represent the original data and ignore the other features, there will exist a probabilistic relation between the original sample and its optimal feature via the indicator as follows:

$$P\left(\boldsymbol{x}_k^{(t)}\right) = P\left(D^{(t)} \mid I_k^{(t)} = 1\right). \tag{2}$$

Obviously, the answer to the aforementioned question would always be available if such indicators were known. In practice, however, the indicators are unknown so that we cannot achieve the optimal performance in this manner. More likely, there is no unique feature highly superior to others for representing all the input samples in $\mathscr{X}$. Therefore, we suggest that all the achieved features are simultaneously used to represent the original samples via indicator variables. For doing so, we specify a finite mixture model as

$$P\left(D^{(t)}\right) = \sum_{k=1}^{K} P\left(D^{(t)} \mid I_k^{(t)} = 1\right) P\left(I_k^{(t)} = 1\right). \tag{3}$$

For this idea, an open problem is how to implement the finite mixture model in Eq. (3) for classification. In the sequel, we shall propose an alternative mixture of experts architecture to solve the problem.

2.2. Alternative mixture of experts architecture

As illustrated in Fig. 1, the *alternative mixture of experts* (AME) architecture is composed of N expert networks and a gate-bank. The ensemble of expert networks is divided into K groups in terms of K diverse features, and there are N_i expert networks in the ith group subject to $\sum_{i=1}^{K} N_i = N$. Expert networks in the same group receive the same feature vector, while any two expert networks in different groups receive different feature vectors. For an input sample, each expert network produces an output vector in terms of a specific feature. In the gate-bank, there are K gating networks and K different feature vectors are input to these networks, respectively. Each gating network produces an output vector in terms of a specific input feature. The output vector consists of N components, where each component corresponds to an expert network. The overall

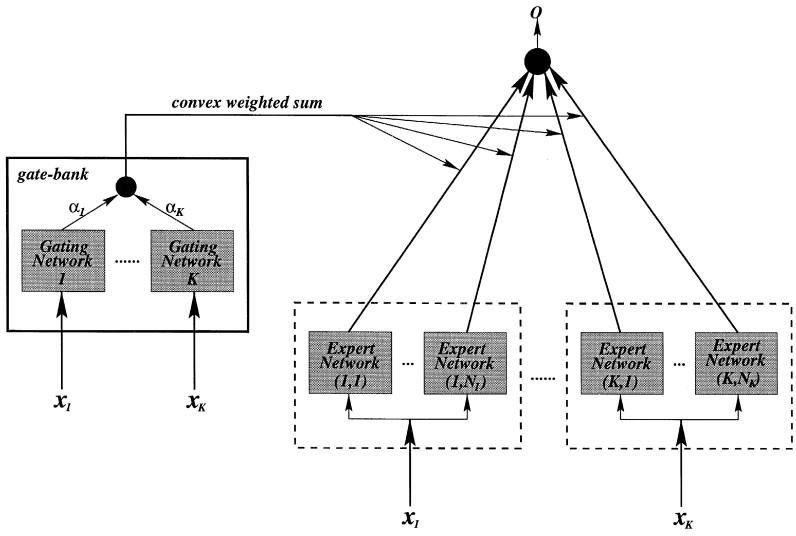

Fig. 1. The alternative mixture of experts architecture. Assume that there are K diverse feature sets extracted from a raw data set. The gate-bank is in the solid-line box, where K gating networks are employed to work on diverse feature sets. Accordingly, the ensemble of expert networks is also divided into K groups. Each group of expert networks in a dash-line box receive the same input.

output of the gate-bank is a convex weighted sum of outputs produced by all the gating networks and can be interpreted as a partition of unity at each point in the input space based on diverse features. As a result, the overall output of the AME architecture is a linear combination of outputs of all N expert networks weighted by the output of the gate-bank.

We stipulate that all the expert networks in the ith group receive the ith feature vector $\boldsymbol{x}_i^{(t)}$ for a sample $D^{(t)}$. Let expert (i,j) denote the jth expert in the ith group. The output of expert (i,j) is

$$\boldsymbol{o}_{ij}\left(\boldsymbol{x}_i^{(t)}\right) = f\left(W_{ij}\boldsymbol{x}_i^{(t)}\right), \tag{4}$$

where W_{ij} is a weight matrix and $f(\cdot)$ is a fixed continuous nonlinear function determined by the statistical model of an expert network. Accordingly, the scalar output of the kth gating network for expert (i,j), a component of the gating network's output vector, is

$$g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}\right) = \frac{e^{\xi_{k,ij}^{(t)}}}{\sum_{u=1}^{K}\sum_{v=1}^{N_i} e^{\xi_{k,uv}^{(t)}}}, \tag{5}$$

where $\boldsymbol{v}_{k,ij}$ is a weight vector in the kth gating network and $\xi_{k,ij}^{(t)} = \boldsymbol{v}_{k,ij}^{\mathrm{T}}\boldsymbol{x}_k^{(t)}$. Furthermore, the corresponding scalar output of the gate-bank for expert network (i,j) is

$$\lambda_{ij}^{(t)} = \sum_{k=1}^{K} \alpha_k\, g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}\right), \tag{6}$$

where α_k is a linear coefficient and satisfies $\sum_{k=1}^{K}\alpha_k = 1$ and $\alpha_k \geqslant 0$. Therefore, the overall output of the AME architecture for the input $D^{(t)}$ is

$$\boldsymbol{o}\left(D^{(t)}\right) = \sum_{i=1}^{K}\sum_{j=1}^{N_i} \lambda_{ij}^{(t)}\boldsymbol{o}_{ij}\left(\boldsymbol{x}_i^{(t)}\right). \tag{7}$$

To explain the proposed architecture, it is useful to provide a probabilistic perspective for the gate-bank and expert networks. The gate-bank is an implementation of the finite mixture model in Eq. (3). $g(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij})$ is interpreted as the probability that an output $\boldsymbol{y}^{(t)}$ is generated by expert network (i,j) based on the feature vector $\boldsymbol{x}_k^{(t)}$. α_k is interpreted as the probability that $\boldsymbol{x}_k^{(t)}$ is the optimal feature of $D^{(t)}$. Therefore, $\lambda_{ij}^{(t)}$ can be interpreted as the probability that an output $\boldsymbol{y}^{(t)}$ is generated by expert network (i,j) according to all the achieved features. Note that the task of the gate-bank is to select an appropriate expert network to generate $\boldsymbol{y}^{(t)}$ for $D^{(t)}$. Once the selection is performed, resulting in a choice of expert network (i,j), $\boldsymbol{y}^{(t)}$ is assumed to be generated according to its statistical model $P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_i^{(t)},W_{ij})$, where W_{ij} is the set of all the parameters in the statistical model (the weight matrix of expert network (i,j) as stated before). Since such a deterministic selection is usually impossible, once again, a soft competition mechanism is adopted for the optimal use of all the expert networks as suggested by Jordan and Jacobs (1994). Therefore, the statistical model of the AME architecture can be described by a generalized finite mixture model, where the total probability of generating $\boldsymbol{y}^{(t)}$ from $D^{(t)}$ can be specified as

$$P\left(\boldsymbol{y}^{(t)} \mid D^{(t)},\Phi\right) = \sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K} \alpha_k\, g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}\right) P\left(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_i^{(t)},W_{ij}\right), \tag{8}$$

where Φ is the set of all parameters in the model, including the expert network parameters W_{ij} and the gate-bank parameters $\boldsymbol{v}_{k,ij}$ and α_k. In terms of pattern classification, the statistical model of an expert network, $P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_i^{(t)},W_{ij})$, is assumed to be the Bernoulli distribution (Jordan and Jacobs, 1994) in the case of binary classification, and the multinomial distribution (Jordan and Jacobs, 1994) or the generalized Bernoulli distribution (Chen et al., 1996b,c) in the case of multicategory classification.

Here, we emphasize that there are two soft competition mechanisms in the AME architecture; on the basis of the supervised error, expert networks compete for the right to learn the training data, while gating networks associated with diverse features compete for the right to select an appropriate expert network as the winner for generating the output.

2.3. EM learning algorithm

Apparently, parameter estimation in the AME architecture is a maximum likelihood learning problem. In this paper, we adopt an *Expectation-Maximization* (EM) algorithm (Dempster et al., 1977) to solve the problem. To develop the EM algorithm, we introduce an additional set of missing data besides the observable data in $\mathscr{X}$ in order to simplify the likelihood function. The set of missing data is denoted as $\mathscr{I}=\{I_{ij}^{(t)},I_k^{(t)}\}_{t=1}^{T}$, where $i=1,\ldots,K$, $j=1,\ldots,N_i$, and $k=1,\ldots,K$. The indicator variable $I_{ij}^{(t)}$ is defined as

$$I_{ij}^{(t)}=\begin{cases}1 & \text{if } \boldsymbol{y}^{(t)} \text{ is generated from expert } (i,j),\\ 0 & \text{otherwise,}\end{cases} \tag{9}$$

where $I_{ij}^{(t)}$ satisfies $\sum_{i=1}^{K}\sum_{j=1}^{N_i}I_{ij}^{(t)}=1$. The indicator variable $I_k^{(t)}$ is the same as defined in Eq. (1) and, here, interpreted as that the selection can be merely performed by the kth gating network when its input $\boldsymbol{x}_k^{(t)}$ is viewed as the optimal feature of $D_k^{(t)}$. As a result, the set of complete data is achieved as $\mathscr{Y}=\{\mathscr{X},\mathscr{I}\}$. Accordingly, the complete-data likelihood function is achieved as

$$\begin{aligned} l_c(\Phi;\mathscr{Y}) &= \log\prod_{t=1}^{T}\prod_{i=1}^{K}\prod_{j=1}^{N_i}\prod_{k=1}^{K}\left\{\alpha_k\, g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}\right)P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}\right)\right\}^{I_{ij}^{(t)}I_k^{(t)}}\\ &= \sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}I_{ij}^{(t)}I_k^{(t)}\left\{\log\alpha_k+\log g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}\right)+\log P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}\right)\right\}. \end{aligned} \tag{10}$$

Consequently, given the observed data and the current model, the E-step of the EM algorithm is defined by taking the expectation of the complete-data likelihood:

$$E\left[l_c(\Phi;\mathscr{Y})\mid\mathscr{X}\right]=\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}h_{ij}^{(t)}h_k^{(t)}h_{k,ij}^{(t)}\left\{\log\alpha_k+\log g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}\right)+\log P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}\right)\right\}, \tag{11}$$

where $h_{ij}^{(t)}$, $h_k^{(t)}$ and $h_{k,ij}^{(t)}$ are the posterior probabilities (for details see Appendix A) and evaluated by

$$h_{ij}^{(t)}=\frac{\sum_{k=1}^{K}\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}, \tag{12}$$

$$h_k^{(t)}=\frac{\alpha_k^{(s)}\sum_{i=1}^{K}\sum_{j=1}^{N_i}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)} \tag{13}$$

and

$$h_{k,ij}^{(t)}=\frac{\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid \boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}. \tag{14}$$

Here, $\Phi^{(s)} = \{\alpha_k^{(s)}, \boldsymbol{v}_{k,ij}^{(s)}, W_{ij}^{(s)}\}$ is the value of the parameters at the sth iteration. Thus, the M-step reduces the problem of maximizing $E[l_c(\Phi;\mathcal{Y}) \mid \mathcal{X}]$ with respect to the expert network parameters and the gate-bank parameters to the following separate maximization problems:

$$W_{ij}^{(s+1)} = \arg\max_{W_{ij}} \sum_{t=1}^{T} h_{ij}^{(t)} \log P\left(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}^{(t)}, W_{ij}\right), \tag{15}$$

$$V_k^{(s+1)} = \arg\max_{V_k} \sum_{t=1}^{T} \sum_{i=1}^{K} \sum_{j=1}^{N_i} h_{k,ij}^{(t)} \log g\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{v}_{k,ij}\right), \tag{16}$$

where V_k is the matrix consisting of all the weight vectors $\boldsymbol{v}_{k,ij}$ in the kth gating network, and

$$\alpha^{(s+1)} = \arg\max_{\alpha} \sum_{t=1}^{T} \sum_{k=1}^{K} h_k^{(t)} \log \alpha_k \quad \text{s.t.} \quad \sum_{k=1}^{K} \alpha_k = 1,\ \alpha_k \geqslant 0, \tag{17}$$

where $\alpha = (\alpha_1, \ldots, \alpha_k)^{\mathrm{T}}$. Consequently, the first two problems in Eqs. (15) and (16) can be solved by the Newton–Raphson method or its approximation, a faster learning algorithm, proposed in our earlier work (Chen et al., 1996c), while the last problem in Eq. (17) can be analytically solved by

$$\alpha_k^{(s+1)} = \frac{1}{T} \sum_{t=1}^{T} h_k^{(t)}. \tag{18}$$

3. Simulations

Speaker identification is to classify an unlabeled voice token as belonging to one of reference speakers. It is a difficult pattern classification task since a person's voice always changes over time. Extensive studies show that no unique robust feature has been found so far and several spectral features are reported to be useful for speaker identification instead (for reviews of the subject see (Doddington, 1986; Campbell, 1997; Furui, 1997)). Therefore, speaker identification becomes a typical problem of classification with diverse features. Speaker identification systems can be either text-dependent or text-independent. Text-dependent means that the same text is used in training and test. In contrast, any text is allowed to be used in either training or test in a text-independent speaker identification system. We have applied the proposed method in both text-dependent and text-independent speaker identification to evaluate its performance. All simulations were done in a Sun sparc-10 workstation.

Speaker identification is of two salient characteristics in contrast to common pattern classification tasks. On the one hand, a speaker identification system trained on the data recorded in multiple sessions often outperforms another system trained on the data recorded in a single session because features extracted from the data recorded in multiple sessions carry more robust information in general. On the other hand, the performance of a speaker identification system will be often degraded when those data recorded in two different sessions of a long time interval is used for training and test, respectively. As mentioned in the introduction, most of the combination methods adopt a two-stage sequential learning procedure so that two data sets are required; one is the training set used to train individual classifiers in the first stage and the other is the so-called cross-validation set to train the combination scheme in the second stage. As a result, all the aforementioned characteristics were taken into consideration in our simulations. For reasonable comparison, simulations with respect to the proposed method were done in two different ways. One was a comparison between the AME architecture and individual classifiers based on the same training set, and the other was a comparison between the combination methods and the AME architecture trained on a data set consisting of data in both the training and the cross-validation sets.

3.1. Results on text-dependent speaker identification

The acoustic database used in simulations consisted of 10 isolated digits from ''0'' to ''9'' uttered in Mandarin. All utterances were recorded in three different sessions, and the time interval between two adjacent sessions was one month. 20 male speakers were registered in the database, and 200 utterances (10 utterances/speaker) were recorded for each digit in each session. In simulations, utterances recorded in the first session were used as the training data, and utterances recorded in two additional sessions were employed as two testing sets, called TEST-1 and TEST-2, respectively. Moreover, the training data was divided into two sets with the same amount of utterances. They were used as the training set and the cross-validation set, respectively. In simulations, we adopted four common speech spectral features widely used in text-dependent speaker identification (Doddington, 1986; Campbell, 1997; Furui, 1997), i.e., 17-order delta-cepstrum, 17-order LPC based cepstrum, 17-order Mel-scale cepstrum, and 13-order LPC coefficients.

Corresponding to 10 isolated digits, 10 AME classifiers were employed in simulations. The generalized Bernoulli distribution was used as the statistical model of each expert network (Chen et al., 1996b,c). The structure was chosen from four AME candidates ranging from 12 to 24 expert networks using the two-fold cross-validation method. Finally, an AME with 16 expert networks was adopted as the structure of the AME classifiers and the ensemble of expert networks in this structure was divided into four groups (four expert networks in each group, see also Fig. 1). The proposed EM algorithm was used to train each AME classifier. In simulations, the AME classifiers were trained on the training set and a larger data set consisting of all the utterances recorded in the first session, respectively. Accordingly, testing results are illustrated in Figs. 2 and 3, respectively, and the CPU time of training those AME classifiers is shown in Figs. 4 and 5, respectively, in terms of two different training sets.

Our earlier work showed that the *hierarchical mixtures of experts* (HME) architecture (Jordan and Jacobs, 1994), a variant of the mixture of experts model, outperforms the ME architecture in speaker identification (Chen et al., 1996a,b). For comparison, we also used 40 HME classifiers to deal with the same problem and each HME classifier was used to handle the utterances of a digit based on a specific feature set. Model selection was also performed by the two-fold cross-validation method. In simulations, seven HMEs with different structures ranging from two to four levels were examined and a three-level HME with 16 expert networks was finally chosen as the structure of HME classifiers. The generalized Bernoulli distribution was also used as the

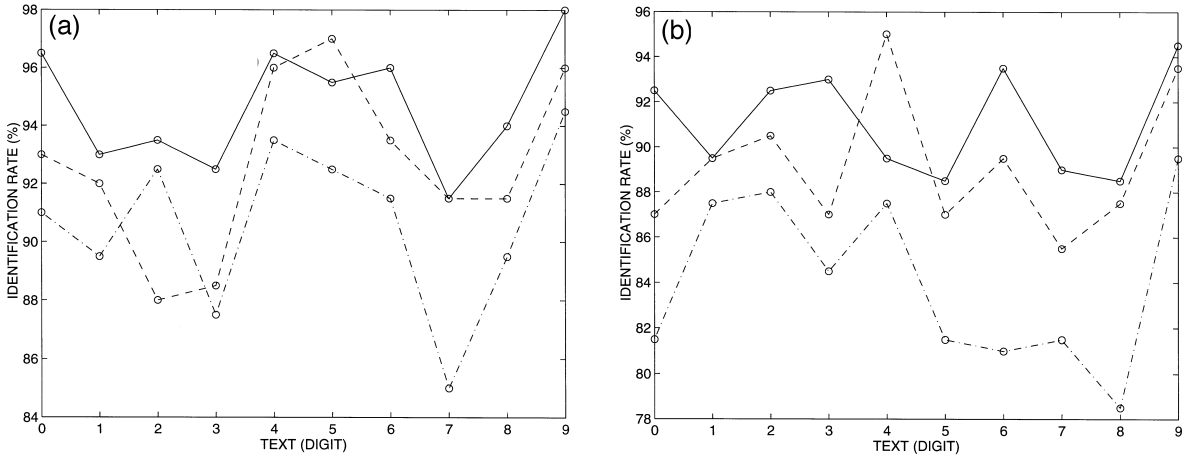

Fig. 2. Identification rates produced by our architecture trained on different feature sets (solid line) and the HME classifiers trained on either individual feature sets (dash-dot line) or the composite feature set (dashed line) in text-dependent speaker identification. All the architectures are trained on the training set. (a) Testing results on TEST-1. (b) Testing results on TEST-2.

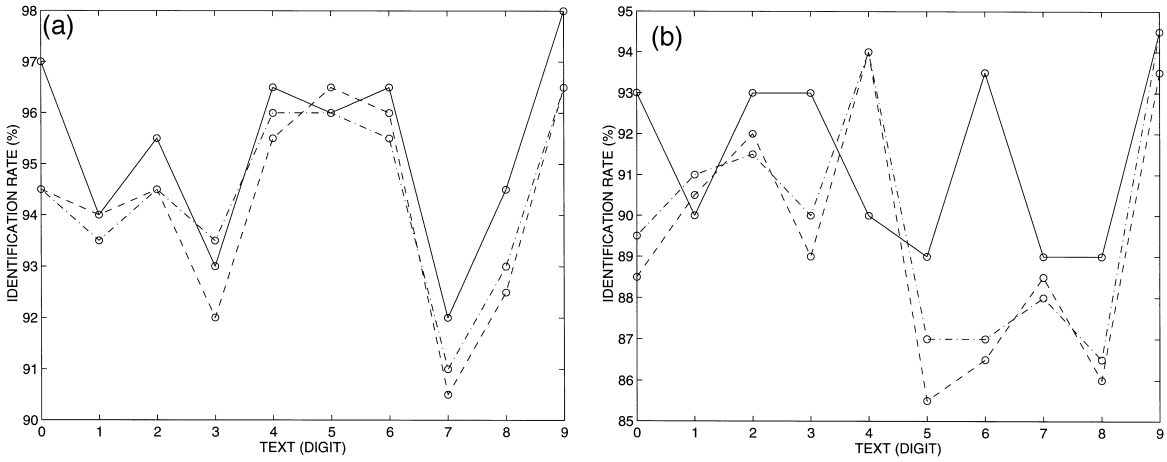

Fig. 3. Identification rates produced by our architecture (solid line), the BAYES combination method (dashed line), and the BKS combination method (dash-dot line) in text-dependent speaker identification. Both our architecture and two combination schemes are trained on the same data set consisting of both the training and the cross-validation sets. (a) Testing results on TEST-1. (b) Testing results on TEST-2.

statistical model of each expert network. The EM algorithm proposed by Jordan and Jacobs (1994) was used for training those HME classifiers on the training set. Due to the limited space here, we merely report the mean identification rates produced by those HME classifiers trained on the four individual feature sets and their mean training time. For the purpose of comparison, mean identification rates produced by those HME classifiers are shown in Fig. 2 and the mean CPU time of training those HME classifiers is shown in Fig. 4. On the other hand, composite-feature based methods are often used for classification with diverse features. For comparison, we also conducted simulations using a composite-feature based method. In simulations, we lumped the aforementioned four different feature vectors together to form a 64-dimensional composite feature vector. 10 aforementioned HME classifiers were trained on the composite feature set generated from the training set. Identification rates produced by those HME classifiers trained on composite feature sets are also illustrated in Fig. 2, and the CPU time of training those HME classifiers on the composite feature set is illustrated in Fig. 4.

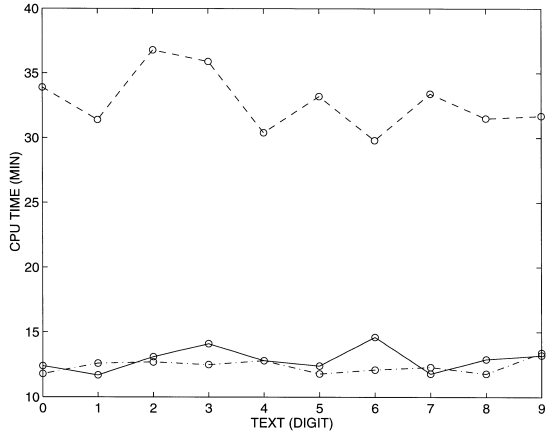

Fig. 4. CPU time of training our architecture on different feature sets (solid line) and the HME classifiers on either individual feature sets (dash-dot line) or the composite feature (dashed line) in text-dependent speaker identification.

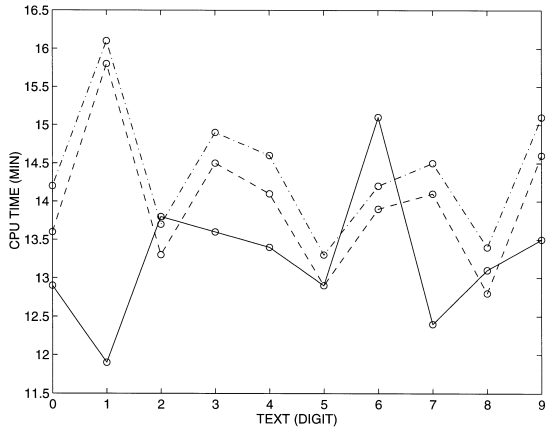

Fig. 5. CPU time of training our architecture (solid line), the BAYES combination method (dashed line), and the BKS combination method (dash-dot line) in text-dependent speaker identification.

Combination of multiple classifiers turns out a good way to handle a task of classification with diverse features. For comparison, we also applied two recent combination methods in the same problem. They are the Bayesian reasoning (BAYES) method (Xu et al., 1992) and the behavior-knowledge space (BKS) method (Huang and Suen, 1995), respectively. It has been reported that the BKS method achieves a promising performance and outperforms several classical combination methods in an unconstrained handwritten numerals recognition problem (Huang and Suen, 1995), while the BAYES method also readily yields good performance in speaker identification (Chen et al., 1997) and the hand-written optical character recognition (Xu et al., 1992; Perrone, 1993). In simulations, the aforementioned HME classifiers trained on different feature sets were used as individual classifiers. The cross-validation set was used to calculate the confusion matrix in the BAYES method (Xu et al., 1992) and to acquire the behavior-knowledge space in the BKS method (Huang and Suen, 1995). Identification rates produced by the two combination methods are illustrated in Fig. 3. For comparison in training time, we also show CPU time of training the two combination schemes in Fig. 5. Note that the training time of a combination scheme is the sum of the mean training time taken for training all the individual HMEs and the time for calculating the confusion matrix or acquiring the behavior-knowledge space.

It is evident from Fig. 2 that the AME classifiers trained on different feature sets produce considerably better performance than that of the HME classifiers trained on either individual feature sets or the composite feature set. In particular, the AME architecture yields significantly faster training than the HME architecture trained on the composite feature set. On the other hand, simulation results illustrated in Fig. 3 also show that our method performs well; its performance is slightly better than that of the two combination method in TEST-1, but it outperforms the two combination method in TEST-2. In addition, the performance of the AME architecture is only slightly degraded when less data is used for training, which indicates that the AME architecture is more robust in text-dependent speaker identification.

3.2. Results on text-independent speaker identification

Text-independent speaker identification is a more difficult learning task. Because the text may be arbitrary at any time, all the template matching techniques are not applicable. Therefore, speaker's features play a critical role in text-independent speaker identification. In order to further evaluate its performance, we have also applied the AME architecture in a text-independent speaker identification task.

The database used in simulations is a subset of the standard Chinese speech database. This set represents 20 speakers of the same Mandarin dialect. The utterances in the database were recorded during three separate

Table 1
Testing results (%) on SET-2 produced by our architecture (AME) trained on different feature sets and HMEs trained on either individual feature sets or the composite feature set in text-independent speaker identification. SET-3 is the training set in simulations

Classifier (feature)	Identification	Substitution	Rejection
AME (diverse features)	89.3	7.6	3.1
HME (19-LPCCEP)	82.8	13.7	3.5
HME (19-MELCEP)	83.6	11.8	4.6
HME (19-LPCCOE)	80.3	16.6	3.1
HME (composite feature)	87.2	8.4	4.4

sessions, and the time interval between two consecutive sessions was two weeks. In the first session, 10 different phonetically rich sentences were uttered by each speaker. The average length of the sentences was about 4.5 seconds. In the two additional sessions, five different sentences were uttered by each speaker, respectively. The average lengths of the sentences recorded in the two sessions were about 4.4 and 5.0 seconds, respectively. In simulations, the data recorded in the first session was used as the training set, called SET-1, and the data recorded in the two additional sessions, called SET-2 and SET-3 respectively, were used as either a testing set or a cross-validation set. As a result, there were 10057 frames in SET-1, 4270 frames in SET-2, and 4604 frames in SET-3. We adopted three common speech spectral features extensively used for text-independent speaker identification (Doddington, 1986; Campbell, 1997; Furui, 1997), i.e., 19-order LPC based cepstrum (19-LPCCEP), 19-order Mel-scale cepstrum (19-MELCEP), and 19-order LPC coefficients (19-LPCCOE).

The evaluation method used is briefly described as follows. The sequence of feature vectors corresponding to testing data is denoted as $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_T\}$. The sequence can be divided into overlapping segments of S feature vectors. The first two segments from a sequence

$$\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_S, \boldsymbol{f}_{S+1}, \boldsymbol{f}_{S+2}, \ldots, \boldsymbol{f}_T$$

would be

$$\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_S \quad \text{and} \quad \boldsymbol{f}_2, \ldots, \boldsymbol{f}_S, \boldsymbol{f}_{S+1}.$$

A test segment length of L seconds would correspond to S feature vectors for an L/S msec frame rate. We used the 6.4 msec frame and $S = 100$ in simulations. Using a segment, the system produces either an identifying result or a rejection. In the ''segment test'' method, an unknown speaker can be identified only if at least 50% input vectors in the segment report the same identifying results; otherwise, the system rejects the unknown speaker. The above steps are repeated for test utterances from each speaker of the population. The final performance evaluation is then computed according to identifying, substitution and rejection rates as defined by Xu et al. (1992).

The two-fold cross-validation method was still used for model selection. As a result, the AME with 24 expert networks located in three groups (eight experts in each group, also see Fig. 1) was finally adopted, and the

Table 2
Testing results (%) on SET-3 produced by our architecture (AME) trained on different feature sets and HMEs trained on either individual feature sets or the composite feature sets in text-independent speaker identification. SET-2 is the training set in simulations

Classifier (feature)	Identification	Substitution	Rejection
AME (diverse features)	93.4	2.8	3.8
HME (19-LPCCEP)	89.5	5.2	5.3
HME (19-MELCEP)	91.1	3.1	5.8
HME (19-LPCCOE)	86.9	5.7	7.4
HME (composite feature)	92.1	3.3	4.6

Table 3
Testing results (%) on SET-2 produced by our architecture (AME) trained on SET-13, the Bayesian combination method (BAYES), and the behavior-knowledge space (BKS) combination method in text-independent speaker identification. SET-3 is used as the training set of the two combination methods in simulations

Classification method	Identification	Substitution	Rejection
AME (diverse features)	89.8	7.1	3.1
BAYES COMBINATION	88.1	7.2	4.7
BKS COMBINATION	87.7	6.7	5.6

generalized Bernoulli distribution was used as the statistical model of expert networks. We used the proposed EM algorithm to train the AME classifier on SET-1. The testing results on SET-2 and SET-3 are shown in Tables 1 and 2, respectively. For comparison with the combination methods, we also used all the sentences in SET-1 and two sentences in either SET-2 or SET-3 to established two new training sets, called SET-12 (5894 frames) and SET-13 (5982 frames), respectively. Once the AME classifiers was trained on SET-12, SET-3 would be the test set, while SET-2 would be used as the test set if the AME classifiers was trained on SET-13. Testing results produced by the AME classifier trained on SET-12 and SET-13 are shown in Tables 3 and 4, respectively.

For comparison, a three-level HME structure with 24 expert networks was employed as the structure of individual classifiers for the same problem. We trained the HME classifiers on three individual feature sets, respectively, and a composite feature set formed by lumping the three different feature vectors together. The generalized Bernoulli distribution was also used as the statistical model of each expert network, and the EM algorithm proposed by Jordan and Jacobs (1994) was employed to train these HME classifiers on SET-1. Testing results on SET-2 and SET-3 are shown in Tables 1 and 2, respectively. In addition, we also applied the two combination methods in the text-independent speaker identification problem by combining three HME classifiers trained on individual feature sets. When SET-3 was used as the test set, SET-2 would be the cross-validation set, and vice versa. Testing results on SET-2 and SET-3 are also shown in Tables 3 and 4, respectively. For the purpose of comparison in training time, we also show the training time of all the classifiers used for the text-independent speaker identification problem in Fig. 6. Note that the training time of a combination method is still the sum of the mean training time of individual classifiers and the mean time for either calculating the confusion matrix or acquiring behavior-knowledge space on either SET-2 or SET-3. In addition, the mean training time of the AME classifier on two different training sets is merely illustrated in Fig. 6.

In summary, simulation results show that the AME classifier outperforms the HME classifiers trained on either individual feature sets or the composite feature set. In particular, the AME architecture yields significantly faster training than the HME architecture trained on the composite feature set. On the other hand, the performance of our method is similar to that of the two combination methods in general. But the combination methods need to use more data for training, and the simulation results also indicate that their performance seems

Table 4
Testing results (%) on SET-3 produced by our architecture (AME) trained on SET-12, the Bayesian combination method (BAYES), and the behavior-knowledge space (BKS) combination method in text-independent speaker identification. SET-2 is used as the training set of the two combination methods in simulations

Classification method	Identification	Substitution	Rejection
AME (diverse features)	94.5	2.3	3.2
BAYES COMBINATION	94.1	3.1	2.8
BKS COMBINATION	94.7	3.6	1.7

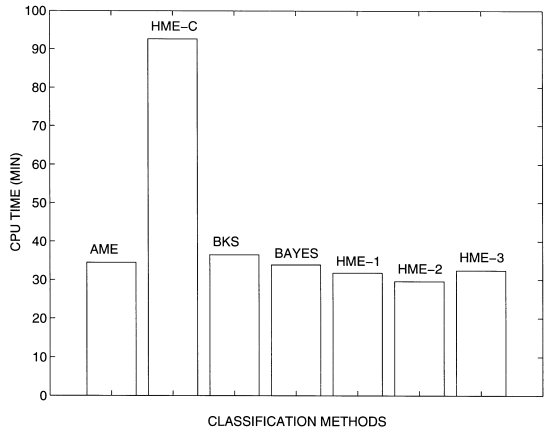

Fig. 6. CPU time of training our architecture (AME), the HME classifier trained on the composite feature set (HME-C), the BKS combination method (BKS), the Bayesian combination method (BAYES), and the HME classifiers trained on individual feature sets in text-independent speaker identification. HME-1,2,3 denote the HME classifiers trained on the 19-LPCCEP feature set, the 19-MELCEP feature set, and the 19-LPCCOE feature set, respectively.

to highly depend upon the cross-validation set used to calculate the confusion matrix and to acquire the behavior-knowledge space. In contrast, our method is robust and performs better when less data is used for training.

4. Conclusions

We have described an alternative method to simultaneously use diverse features in an optimal way for pattern classification. Simulation results have shown that the proposed connectionist method yields the improved performance and fast training. Comparative results also show that the proposed modular neural network architecture outperforms the hierarchical mixture of experts architecture trained on either individual feature sets or a composite feature set and are more robust in comparison with the methods of combining multiple classifiers in speaker identification. In addition, the proposed architecture can be viewed as an extension of the original mixture experts architecture (Jacobs et al., 1991) for classification with diverse features. When a single feature set is used, our architecture will be equivalent to the mixture experts architecture. As an extension of the mixture of experts model, the hierarchical mixtures of experts architecture has shown its effectiveness in many complicated supervised learning tasks. Similarly, the proposed modular neural network architecture can be also extended to a hierarchical structure for classification with diverse features (Chen and Chi, 1996). We expect that such a hierarchical structure yields improved performance in complicated pattern classification tasks.

As a new method for classification with diverse features, our method adopts a single stage learning process rather than a two-stage sequential learning procedure used in methods of combining multiple classifiers. For most of the combination methods, reliable combination schemes need training on a cross-validation data set by almost exhaustive enumeration. As pointed out by Ho et al. (1994), in general, n^k combinations need to be covered by the training data to sufficient density for n classes and k classifier, and therefore computation in those methods is expensive. Although some ad hoc methods have been explored to constrain the combinations based on correlation of classifiers, a systematic approach is still a challenging open problem (Ho et al., 1994). In contrast, such a cross-validation data set is not necessary to be used in our method, and simulations have shown that less training data is required in our method to achieve a similar performance. These salient features significantly distinguish our method from those methods of combining multiple classifiers.

However, model selection is still inevitable in our method. In our simulations, we adopted the time-consuming cross-validation method for model selection. Apparently, more efficient model selection techniques are worth studying for the proposed architecture. The state-of-the-art statistical learning theory provides a feasible way for model selection in general. In our future work, we are going to utilize Bayesian learning and regularization techniques to develop an efficient model selection method for the proposed architecture.

Acknowledgements

The author would like to thank Professors L. Xu and H. Chi for valuable discussions as well as Professor E. Gelsema and two anonymous reviewers for their extensive comments that have significantly improved the presentation of this paper. This work was supported in part by Chinese National Science Foundation under grants 69571002 and 69635020 as well as the NSF grant IRI-9423312.

Appendix A

In this appendix, we present the derivation of the E-step in the proposed EM learning algorithm described in Section 2.3. In the E-step, posterior probabilities $h_{ij}^{(t)}$, $h_k^{(t)}$ and $h_{k,ij}^{(t)}$ are defined as

$$h_{ij}^{(t)} = E\left[I_{ij}^{(t)} \mid \mathscr{X} \right], \quad h_k^{(t)} = E\left[I_k^{(t)} \mid \mathscr{X} \right] \quad \text{and} \quad h_{k,ij}^{(t)} = E\left[I_{ij}^{(t)}, I_k^{(t)} \mid \mathscr{X} \right].$$

$E[I_{ij}^{(t)} \mid \mathscr{X}]$ is computed using the Bayesian rule as

$$E\left[I_{ij}^{(t)} \mid \mathscr{X} \right] = P\left(I_{ij}^{(t)} = 1 \mid \boldsymbol{y}^{(t)}, D^{(t)}, \Phi^{(s)} \right) = \frac{P\left(\boldsymbol{y}^{(t)} \mid I_{ij}^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right) P\left(I_{ij}^{(t)} = 1 \mid D^{(t)}, \Phi^{(s)} \right)}{P\left(\boldsymbol{y}^{(t)} \mid D^{(t)}, \Phi^{(s)} \right)}. \tag{19}$$

According to the total probability rule, furthermore, $P(I_{ij}^{(t)} = 1 \mid D^{(t)}, \Phi^{(s)})$ is computed as

$$P\left(I_{ij}^{(t)} = 1 \mid D^{(t)}, \Phi^{(s)} \right) = \sum_{k=1}^{K} P\left(I_k^{(t)} = 1 \mid D^{(t)}, \Phi^{(s)} \right) P\left(I_{ij}^{(t)} = 1 \mid I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right). \tag{20}$$

According to definitions, we have $P(I_{ij}^{(t)} = 1 \mid I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = g(\boldsymbol{x}_k^{(t)}, \boldsymbol{v}_{k,ij}^{(s)})$, $P(I_k^{(t)} = 1 \mid D^{(t)}, \Phi^{(s)}) = \alpha_k^{(s)}$ and $P(\boldsymbol{y}^{(t)} \mid I_{ij}^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_i^{(t)}, W_{ij}^{(s)})$. $P(\boldsymbol{y}^{(t)} \mid D^{(t)}, \Phi^{(s)})$ is the statistical model of the AME architecture as defined in Eq. (8). Therefore, inserting Eq. (20) into Eq. (19) yields

$$E\left[I_{ij}^{(t)} \mid \mathscr{X} \right] = \frac{\sum_{k=1}^{K} \alpha_k^{(s)} g\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{v}_{k,ij}^{(s)} \right) P\left(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_i^{(t)}, W_{ij}^{(s)} \right)}{\sum_{i=1}^{K} \sum_{j=1}^{N_i} \sum_{k=1}^{K} \alpha_k^{(s)} g\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{v}_{k,ij}^{(s)} \right) P\left(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}_i^{(t)}, W_{ij}^{(s)} \right)}. \tag{21}$$

Using the Bayesian rule, we also have

$$E\left[I_k^{(t)} \mid \mathscr{X} \right] = P\left(I_k^{(t)} = 1 \mid \boldsymbol{y}^{(t)}, D^{(t)}, \Phi^{(s)} \right) = \frac{P\left(\boldsymbol{y}^{(t)} \mid I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right) P\left(I_k^{(t)} = 1 \mid D^{(t)}, \Phi^{(s)} \right)}{P\left(\boldsymbol{y}^{(t)} \mid D^{(t)}, \Phi^{(s)} \right)}. \tag{22}$$

According to the total probability rule,

$$\begin{aligned} P\left(\boldsymbol{y}^{(t)} \mid I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right) &= \sum_{i=1}^{K} \sum_{j=1}^{N_i} P\left(\boldsymbol{y}^{(t)} \mid I_k^{(t)} = 1, I_{ij}^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right) P\left(I_{ij}^{(t)} = 1 \mid I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right) \\ &= \sum_{i=1}^{K} \sum_{j=1}^{N_i} P\left(\boldsymbol{y}^{(t)} \mid I_{ij}^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right) P\left(I_{ij}^{(t)} = 1 \mid I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)} \right). \end{aligned} \tag{23}$$

Note that the indicator variable $I_k^{(t)}=1$ can be ignored from $P(\boldsymbol{y}^{(t)}\mid I_k^{(t)}=1,I_{ij}^{(t)}=1,D^{(t)},\Phi^{(s)})$ in Eq. (23) since it is independent of the probability model based on the fact that $\boldsymbol{y}^{(t)}$ is generated from expert network (i,j) regardless of any gating network in the gate-bank. Therefore, inserting Eq. (23) into Eq. (22) results in

$$E\left[I_k^{(t)}\mid\mathscr{X}\right]=\frac{\alpha_k^{(s)}\sum_{i=1}^{K}\sum_{j=1}^{N_i}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid\boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid\boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}. \tag{24}$$

Similarly,

$$\begin{aligned}E\left[I_{ij}^{(t)},I_k^{(t)}\mid\mathscr{X}\right]&=P\left(I_{ij}^{(t)}=1,I_k^{(t)}=1\mid\boldsymbol{y}^{(t)},D^{(t)},\Phi^{(s)}\right)\\&=\frac{P\left(\boldsymbol{y}^{(t)}\mid I_{ij}^{(t)}=1,I_k^{(t)}=1,D^{(t)},\Phi^{(s)}\right)P\left(I_{ij}^{(t)}=1,I_k^{(t)}=1\mid D^{(t)},\Phi^{(s)}\right)}{P\left(\boldsymbol{y}^{(t)}\mid D^{(t)},\Phi^{(s)}\right)}\\&=\frac{P\left(\boldsymbol{y}^{(t)}\mid I_{ij}^{(t)}=1,D^{(t)},\Phi^{(s)}\right)P\left(I_{ij}^{(t)}=1,I_k^{(t)}=1\mid D^{(t)},\Phi^{(s)}\right)}{P\left(\boldsymbol{y}^{(t)}\mid D^{(t)},\Phi^{(s)}\right)}\end{aligned} \tag{25}$$

and

$$P\left(I_{ij}^{(t)}=1,I_k^{(t)}=1\mid D^{(t)},\Phi^{(s)}\right)=P\left(I_{ij}^{(t)}=1\mid I_k^{(t)}=1,D^{(t)},\Phi^{(s)}\right)P\left(I_k^{(t)}=1\mid D^{(t)},\Phi^{(s)}\right). \tag{26}$$

Assembling Eq. (25) and Eq. (26), we achieve

$$E\left[I_{ij}^{(t)},I_k^{(t)}\mid\mathscr{X}\right]=\frac{\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid\boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k^{(s)}g\left(\boldsymbol{x}_k^{(t)},\boldsymbol{v}_{k,ij}^{(s)}\right)P\left(\boldsymbol{y}^{(t)}\mid\boldsymbol{x}_i^{(t)},W_{ij}^{(s)}\right)}. \tag{27}$$

References

Campbell, F.P., 1997. Speaker recognition: A tutorial. Proc. IEEE 85 (9), 1437–1463.

Chen, K., Chi, H., 1996. A modular neural network architecture for pattern classification with diverse features. Technical Report PKU-CIS-96-TR07. National Laboratory of Machine Perception and Center for Information Science, Peking University.

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

Chen, K., Xie, D., Chi, H., 1996b. Speaker identification using time-delay HMEs. Internat. J. Neural Syst. 7 (1), 29–43.

Chen, K., Xu, L., Chi, H., 1996c. Improved learning algorithms for mixtures of experts in multiclass classification, Neural Networks (in revision).

Chen, K., Wang, L., Chi, H., 1997. Method of combining multiple classifiers with different features and their applications to text-independent speaker recognition. Internat. J. Pattern Recogn. Artif. Intell. 11 (3), 417–445.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum-likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. Ser. B 39 (1), 1–38.

Doddington, G., 1986. Speaker recognition – identifying people by their voice. Proc. IEEE 73 (11), 1651–1664.

Fogelman-Soulie, F., Viennet, E., Lamy, B., 1993. Multi-modular neural network architectures: applications in optical character and human face recognition. Internat. J. Recogn. Artif. Intell. 7 (4), 721–755.

Furui, S., 1997. Recent advances in speaker recognition. Pattern Recognition Letters 18 (9), 859–872.

Ho, T., Hull, J., Srihari, S., 1994. Decision combination in multiple classifier systems. IEEE Trans. Pattern Anal. Machine Intell. 16 (1), 66–75.

Huang, Y., Suen, C., 1995. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. IEEE Trans. Pattern Anal. Machine Intell. 17 (1), 90–94.

Jacobs, R., Jordan, M., Nowlan, S., Hinton, G., 1991. Adaptive mixture of local experts. Neural Computation 3 (1), 79–87.

Jordan, M., Jacobs, R., 1994. Hierarchical mixtures of experts and the EM algorithm. Neural Computations 6 (3), 181–214.

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

Suen, C.Y., Legault, R., Nadal, C., Cheriet, M., Lam, L., 1993. Building a new generation of handwriting recognition systems. Pattern Recognition Letters 14 (4), 303–315.

Xu, L., Krzyzak, A., Suen, C., 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. System Man Cybernet. 23 (3), 418–435.