International Journal of Neural Systems, Vol. 9, No. 6 (December, 1999) 563–581

A MODULAR NEURAL NETWORK ARCHITECTURE FOR PATTERN CLASSIFICATION BASED ON DIFFERENT FEATURE SETS

KE CHEN* and HUISHENG CHI
National Laboratory of Machine Perception and Center for Information Science, Peking University, Beijing 100871, China

Received 25 April 1996
Revised 30 July 1997
Accepted 20 March 1999

We propose a novel connectionist method for the use of different feature sets in pattern classification. Unlike traditional methods, e.g., combination of multiple classifiers and use of a composite feature set, our method copes with the problem based on an idea of soft competition on different feature sets developed in our earlier work. An alternative modular neural network architecture is proposed to provide a more effective implementation of soft competition on different feature sets. The proposed architecture is interpreted as a generalized finite mixture model and, therefore, parameter estimation is treated as a maximum likelihood problem. An EM algorithm is derived for parameter estimation and, moreover, a model selection method is proposed to fit the proposed architecture to a specific problem. Comparative results are presented for the real world problem of speaker identification.

1. Introduction

The problem of pattern classification can be stated as follows: Given a set of training data D, each with an associated label $\mathbf{y}$, find a classification system that will produce the correct label $\mathbf{y}$ for any data D drawn from the same source as the training data. In general, a typical pattern classification system, as depicted in Fig. 1(a), is composed of three stages: preprocessing, feature extraction, and classification. For real world problems, both preprocessing and feature extraction are necessary prior to training of a classification system in order to avoid the curse of dimensionality.[11] Therefore, the performance of a classification system highly depends upon a feature set used. For a complicated pattern classification task, there are often a number of methods available for feature extraction. By these methods, a raw data set is represented by several different feature sets, which leads to a problem how to utilize those feature sets for classification. To our knowledge, there are two frameworks to tackle the problem; one is the use of feature selection to achieve an optimal feature set,[2] and the other is the joint use of different feature sets. If an optimal feature set can be achieved for a raw data set, we would merely use it to train a classification system, as shown in Fig. 1(a). However, such an optimal set is not achieved often. In this circumstance, the individual use of different feature sets leads to similar performance in classification and, as depicted in Fig. 1(b), the joint use of different feature sets results in better performance or a robust effect. In the real world, there are many such instances, such as speaker recognition[10,15,23,26] and hand-written optical character recognition (OCR)[1,31,32] etc. In this paper, we call such a kind of pattern classification tasks *pattern classification based on different feature sets*.

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

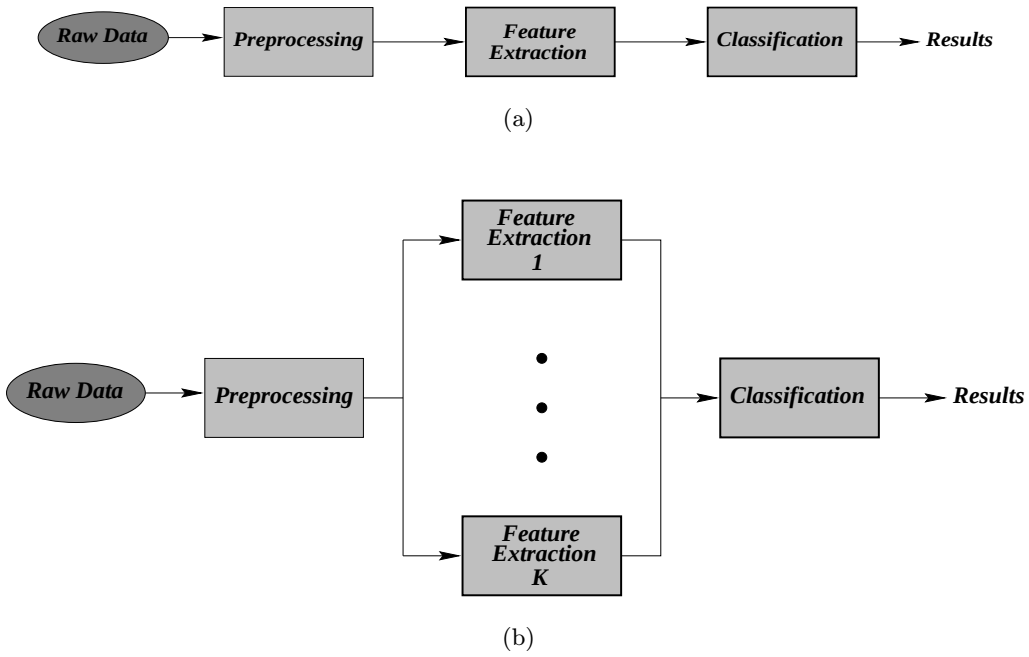


Fig. 1. Typical pattern classification systems. (a) A system based on one feature set. (b) A system based on different feature sets.

There have been two traditional methods to simultaneously use different feature sets for classification; i.e., use of a composite feature set and combination of multiple classifiers trained on different feature sets. In a composite feature set, a composite feature vector is generated by lumping several different feature vectors together. The basic idea behind the use of a composite feature set is to consider an integration of different feature sets as a single feature set that can represent a raw data better than one of components. Although the use of a composite feature set may improve performance of classification, the following problems are unavoidable: (a) Curse of dimensionality; the dimension of a composite feature vector may be much higher than any of component feature vectors. (b) Difficulty in formation; it may be difficult to lump several different feature vectors together due to their diversified forms. (c) Redundancy; the component feature vectors may not be independent of each other. Due to the aforementioned problems, the use of a composite feature set does not result in significant improvements. On the other hand, there have been extensive studies on classification by combining multiple classifiers trained on different feature sets.[4,5,16,21,24,31,32] The basic idea underlying these methods is somehow to learn from outputs of the multiple classifiers trained on different feature sets. Basically most of those methods are viewed as applications of a general approach called stacked generalization outlined by Wolpert.[29] Consequently, the learning of a pattern classification task consists of two phases; each classifier is first trained on a training set, and then a combination scheme is trained on a cross-validation set. Recent studies show that combination of multiple classifiers trained on different feature sets results in the significantly improved performance. However, a sufficiently large data set is usually demanded for that two-stage learning. In addition, combination of multiple classifiers is also viewed as a specific hybrid multi-modular architecture for pattern classification. Due to the use of sequential training, such a hybrid multi-modular architecture works in a sub-optimal style.[12]

We have proposed an alternative method that simultaneously utilizes different feature sets for pattern classification.[3,4] The basic idea underlying the method is a soft competition scheme for the optimal use of different feature sets. A critical issue in the alternative method is how to provide an

effective implementation of the soft competition scheme. Previously, we proposed a modular neural network architecture,[3] which is an extension of the mixture-of-expert (ME) model.[17] Although it outperforms two traditional methods, the global soft competition among subnetworks results in unexpected performance; subnetworks trained on different feature sets may not utilize the information from the same feature set sufficiently. In addition, model selection is an open problem for that modular neural network architecture.[3] In this paper, we propose an alternative connectionist implementation for the soft competition scheme. The proposed implementation is a modular neural network architecture, which can be regarded as a generalized mixture-of-expert (GME) model. Unlike the mixture-of-expert architecture,[17] a gate-bank consists of several gating networks trained on different feature sets, while multiple expert-banks are trained on different feature sets and each expert-bank consists of multiple expert networks trained on the same feature set. In our architecture, there are three soft competition schemes; gating networks based on different feature sets compete for the right to stochastically select an appropriate expert-bank as the winner, expert-banks based on different feature sets compete for the right to produce an output, and in each expert-bank expert networks compete for the right to learn the training data in terms of a single feature set. The proposed architecture can be interpreted as a generalized finite mixture model from the viewpoint of statistics. Therefore, learning in this architecture is treated as a maximum likelihood problem and an EM algorithm is derived for adjusting the parameters in our architecture. Motivated by recent work,[18,30] moreover, we propose a model selection method by means of the maximal likelihood and the cross-validation principles to determine an appropriate structure of the GME classifier along with learning for a specific problem. In order to evaluate the proposed architecture, we have applied our architecture to a real world problem, *speaker identification*, in which different feature sets usually need to be jointly used for robustness. Simulation results have demonstrated that the proposed architecture along with the EM algorithm yields satisfactory results and fast training and, moreover, the proposed model selection method leads to better generalization performance. For comparison, we also applied ME classifiers trained on either individual feature sets or a composite feature set and a method of combining multiple classifiers trained on different feature sets to the same problem. Comparative results indicate that our method yields better performance.

The remainder of this paper is organized as follows. Section 2 presents model description. Section 3 presents an EM algorithm for parameter estimation and a model selection method for structure pruning. Section 4 reports simulation results on speaker identification and conclusions are drawn in the last section.

2. Model Description

In this section, we first review the soft competition scheme on different feature sets proposed in our previous work.[3,4] Then, we present a novel modular neural network architecture to provide an alternative implementation of the soft competition scheme. To understand our model better, a probabilistic interpretation of our model is given in this section.

2.1. *Soft competition scheme for the use of 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 be extracted from a raw data set. Thus, K different feature vectors, $\mathbf{x}_1(D^{(t)}), \ldots, \mathbf{x}_K(D^{(t)})$, can be achieved to represent the sample $D^{(t)}$ in diversified forms for an input sample $D^{(t)}$ in a raw data set, $\mathcal{X} = \{D^{(t)}, \mathbf{y}^{(t)}\}_{t=1}^{T}$. To simplify the presentation, hereinafter, we drop the specific sample term, $D^{(t)}$, from those feature vectors as $\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_K^{(t)}$.

Suppose that there is an optimal feature vector which is the best one to represent the corresponding raw datum among K different feature vectors. Thus, a problem can be addressed: which one is the optimal feature vector of the sample, $D^{(t)}$, among its K different feature vectors, $\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_K^{(t)}$? Apparently, a feature selection technique must be used to generate a solution to this problem. Unfortunately, such a method is often not available in many real world problems. It implies that different feature sets can independently represent a raw data set but in general none of them provides much better representation in comparison with others. Due to high complexity of a real world problem, a single feature set represents

only partial data well, and the joint use of different feature sets can represent all the data better. In this circumstance, we have proposed a solution to this problem.[3,4]

Prior to addressing the solution, we first introduce a set of binary indicator variables to represent the optimal feature vector. An indicator, $I_k^{(t)}$, corresponding to feature vector $\mathbf{x}_k^{(t)}$ is defined as $I_k^{(t)} = 1$ if $\mathbf{x}_k^{(t)}$ 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 other feature vectors, there would exist a probabilistic relation between the raw datum and its optimal feature vector via the indicator as follows:

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

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. As pointed out above, it is more likely that there is no unique feature set highly superior to other feature sets to represent all the input samples. Therefore, the basic idea is the joint use of 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. For such a method, an open problem is how to utilize the mixture model to implement the soft competition on different feature sets. In this sequel, we propose a novel modular neural network architecture to solve this problem.

2.2. *Architecture*

As illustrated in Fig. 2, the proposed GME architecture consists of a gate-bank, where there are K gating networks, and K expert-banks assuming that K different feature vectors can be extracted from an

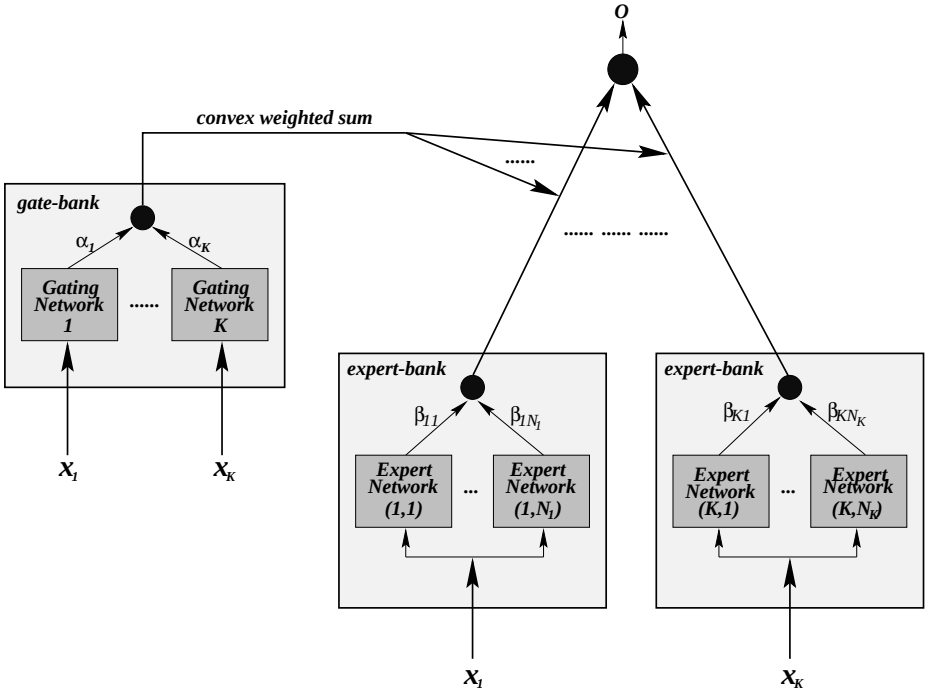


Fig. 2. The generalized mixture-of-expert architecture.

original simple D. Unlike the standard ME model, a gate-bank is used in our architecture to implement the basic idea on the use of different feature sets for classification. The kth gating network in the gate-bank always receives the feature vector $\mathbf{x}_k$ and produces scalar outputs as a partition of unity at each point in the input space based on only the feature vector $\mathbf{x}_k$. By the output of each gating network, the gate-bank produces scalar outputs as a partition of unity at each point in the input space in terms of different feature sets. In contrast to the standard ME model where a number of expert networks are employed, K expert-banks are used to deal with different feature sets, respectively, for classification. The ith expert-bank consists of N_i expert networks which always receive the same feature vector, $\mathbf{x}_i$, while different expert-banks always receive the different feature vectors. The outputs of expert networks in a expert-bank are linearly combined to form an output of this expert-bank. The final output of our architecture is a convex weighted sum of output vectors produced by K expert-banks.

In the GME architecture, each expert network is linear with a single output nonlinearity; i.e., the jth expert network in the ith expert-bank, (i, j), produces its output, $\mathbf{o}_{ij}$, as a generalized linear function of the input $\mathbf{x}_i$:

$$\mathbf{o}_{ij} = f(W_{ij}\mathbf{x}_i)\,, \tag{3}$$

where W_{ij} is a weight matrix and f is a fixed continuous nonlinearity. The vector $\mathbf{x}_i$ is assumed to include a fixed component of one to allow for an intercept term. The output of the ith expert-bank, $\mathbf{o}_i$, is

$$\mathbf{o}_i = \sum_{j=1}^{N_i} \beta_{ij}\mathbf{o}_{ij}\,, \tag{4}$$

where β_{ij} are linear coefficients to combine the outputs produced by expert networks in the ith expert-bank on the conditions: $\sum_{j=1}^{N_i} \beta_{ij} = 1$ and $\beta_{ij} \geq 0$.

The kth gating network in the gate-bank is also generalized linear. As a result, the ith output of the kth gating network, $g_{k,i}$, is the softmax function of intermediate variables $\xi_{k,i}$:

$$g_{k,i} = \frac{e^{\xi_{k,i}}}{\sum_{u=1}^{K} e^{\xi_{k,u}}}\,, \tag{5}$$

where $\xi_{k,i} = \mathbf{v}_{k,i}^T\mathbf{x}_k$ and $\mathbf{v}_{k,i}$ is a weight vector. Furthermore, the ith output of the gate-bank, λ_i, is

$$\lambda_i = \sum_{k=1}^{K} \alpha_k g_{k,i}\,, \tag{6}$$

where α_k are linear coefficients for combining the outputs produced by gating networks in the gate-bank on the conditions: $\sum_{k=1}^{K} \alpha_k = 1$ and $\alpha_k \geq 0$. Therefore, the total output, $\mathbf{o}$, of the GME is

$$\begin{aligned}\mathbf{o} &= \sum_{i=1}^{K} \lambda_i \mathbf{o}_i \\ &= \sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K} \alpha_k g_{k,i}\beta_{ij}\mathbf{o}_{ij}\,.\end{aligned} \tag{7}$$

2.3. *Probabilistic interpretation*

In order to understand our architecture, it is helpful to present a probabilistic interpretation. As a result, this probabilistic interpretation provides a statistical model for the GME architecture in turn so that an efficient learning algorithm can be developed for parameter estimation.

The probabilistic interpretation is described as follows. The gate-bank is an implementation of the finite mixture model in Eq. (2). $g_{k,i}$ is the probability that the ith expert-bank is chosen for classification based on the optimal feature vector $\mathbf{x}_k$ of the sample D, while α_k could be interpreted as the probability that the feature vector $\mathbf{x}_k$ is optimal among K different feature vectors of the sample D accordingly. Thus, λ_i is interpreted as the multinomial probability which can make the decision that terminates in a regressive process that maps D to $\mathbf{y}$. On the other hand, β_{ij} could be interpreted as the probability that $\mathbf{y}$ is generated by the expert network (i, j) when expert-bank i has been chosen to deal with the current input sample for classification. Once the decision has been made by the gate-bank, resulting in a choice of the ith expert-bank, output $\mathbf{y}$ is assumed to be generated according to the statistical model $P(\mathbf{y}|\mathbf{x}_i, \theta_i)$, where θ_i denotes the set of all the parameters in the probabilistic model of the ith expert-bank. The regressive process associated with the ith expert network is described by a finite

mixture model:

$$P(\mathbf{y}|\mathbf{x}_i,\,\theta_i)=\sum_{j=1}^{N_i}\beta_{ij}P(\mathbf{y}|\mathbf{x}_i,\,\theta_{ij})\,, \tag{8}$$

where $P(\mathbf{y}|\mathbf{x}_i,\,\theta_{ij})$ is the statistical model of expert network $(i,\,j)$ and θ_{ij} is the set of all the parameters in the statistical model. Therefore, the total probability of generating $\mathbf{y}$ from D can be viewed as the mixture of the probabilities of generating $\mathbf{y}$ from component densities, in terms of the raw data, through use of soft competition on different feature sets. Thus, the generalized finite mixture model of the GME architecture in the parameter form is

$$\begin{aligned}
&P(\mathbf{y}|D,\,\Phi)\\
&=\sum_{i=1}^{K}\lambda_i P(\mathbf{y}|\mathbf{x}_i,\,\theta_i)\\
&=\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}[\alpha_k g_{k,i}(\mathbf{x}_k,\,\mathbf{v}_{k,i})][\beta_{ij}P(\mathbf{y}|\mathbf{x}_i,\,\theta_{ij})]\\
&=\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k\beta_{ij}g_{k,i}(\mathbf{x}_k,\,\mathbf{v}_{k,i})P(\mathbf{y}|\mathbf{x}_i,\,\theta_{ij})\,,
\end{aligned} \tag{9}$$

where Φ includes all the expert network parameters (θ_{ij} and β_{ij}) as well as the gate-bank parameters ($\mathbf{v}_{k,i}$ and α_k). Since the model is merely used for pattern classification based on different feature sets, the probabilistic component of the model, $P(\mathbf{y}|\mathbf{x}_i,\,\theta_{ij})$, is assumed to be Bernoulli distribution in the case of binary classification,[19] multinomial logit distribution or the generalized Bernoulli distribution[7,9,19] in the case of multiway classification.

Here we mention that the ME architecture could be a case of the GME architecture when an optimal feature set is available by a feature selection method. In this case, only one single network is required in the gate-bank, and each expert-banks will be substituted by a single expert network, where the optimal feature vectors are fed to all the expert networks. Like the hierarchical mixture-of-expert (HME) model,[19] an extension of our model can make the GME model become a hierarchical architecture and will be presented in Appendix 1.

3. Learning Algorithms

In this section, we derive an EM learning algorithm for parameter estimation in the GME architecture and propose a model selection method to generate an appropriate GME structure for a given classification problem during training.

3.1. *EM algorithm*

Suppose that a training set is given as $\mathcal{X}=\{(D^{(t)},\,\mathbf{y}^{(t)}),\;t=1,\ldots,T\}$, where K feature vectors, $\mathbf{x}_1^{(t)},\ldots,\mathbf{x}_K^{(t)}$, are extracted from $D^{(t)}$. All the paired data in $\mathcal{X}$ are called observable data. To develop an EM algorithm for the GME architecture, a set of missing data are introduced to simplify the likelihood function. The set of missing data with binary value are denoted as

$$\begin{aligned}
\mathcal{I}=\{&I_i^{(t)},\,I_{j|i}^{(t)},\,I_k^{(t)};\;i=1,\ldots,K,\\
&j=1,\ldots,N_i,\;k=1,\ldots,K\}\,,
\end{aligned} \tag{10}$$

where the indicator variable $I_i^{(t)}$ is defined as

$$I_i^{(t)}=\begin{cases}1 & \text{if } \mathbf{y}^{(t)} \text{ is generated from the } i\text{th}\\ & \quad\text{expert-bank.}\\ 0 & \text{otherwise}\end{cases} \tag{11}$$

and the indicator variable $I_{j|i}^{(t)}$ is defined as

$$I_{j|i}^{(t)}=\begin{cases}1 & \text{if } \mathbf{y}^{(t)} \text{ is generated from the}\\ & \quad j\text{th expert network in the}\\ & \quad i\text{th expert-bank.}\\ 0 & \text{otherwise}\end{cases} \tag{12}$$

Given $I_i^{(t)}$ and $I_{j|i}^{(t)}$, the indicator variable $I_{ij}^{(t)}$ is the product of $I_i^{(t)}$ and $I_{j|i}^{(t)}$. The indicator variable $I_k^{(t)}$ is defined as

$$I_k^{(t)}=\begin{cases}1 & \text{if the decision is made by}\\ & \quad\text{the } k\text{th gating network in}\\ & \quad\text{the gate-bank.}\\ 0 & \text{otherwise}\end{cases} \tag{13}$$

These indicator variables satisfy the conditions: $\sum_{i=1}^{K}I_i^{(t)}=1$, $\sum_{j=1}^{N_i}I_{j|i}^{(t)}=1$, $\sum_{i=1}^{K}\sum_{j=1}^{N_i}I_{ij}^{(t)}=1$, and $\sum_{k=1}^{K}I_k^{(t)}=1$. Hence the complete data, $\mathcal{Y}$, are composed of both observable and missing data as $\mathcal{Y}=\{\mathcal{X},\,\mathcal{I}\}$.

Note that the dependence of the probabilities $g_{k,i}(\mathbf{x}_k,\,\mathbf{v}_{k,i})$ was explicitly indicated on $\mathbf{x}_k$ and on the parameters in Eq. (9) and expert networks in the ith expert-bank explicitly receive the same input

vector $\mathbf{x}_i$. In the remainder of the paper, we drop the explicit reference to the input and the parameters to simplify the notation. As a result, the probability model in Eq. (9) is rewritten as

$$P(\mathbf{y}|D, \Phi) = \sum_{i=1}^{K} \sum_{j=1}^{N_i} \sum_{k=1}^{K} \alpha_k \beta_{ij} g_{k,i} P_{ij}(\mathbf{y}) . \quad (14)$$

By the complete data, hence, this probability model can be written in terms of $I_{ij}^{(t)}$ and $I_k^{(t)}$ as follows:

$$\begin{aligned} &P(\mathbf{y}^{(t)}, I_{ij}^{(t)}, I_k^{(t)}|D^{(t)}, \Phi) \\ &\quad = \alpha_k \beta_{ij} g_{k,i}^{(t)} P_{ij}(\mathbf{y}^{(t)}) \\ &\quad = \prod_{i=1}^{K} \prod_{j=1}^{N_i} \prod_{k=1}^{K} \{\alpha_k \beta_{ij} g_{k,i}^{(t)} P_{ij}(\mathbf{y}^{(t)})\}^{I_{ij}^{(t)} I_k^{(t)}} . \end{aligned} \quad (15)$$

Taking the logarithm of this probability model yields the following complete-data likelihood:

$$\begin{aligned} &l_c(\Phi; \mathcal{Y}) \\ &\quad = \sum_{t=1}^{T} \sum_{i=1}^{K} \sum_{j=1}^{N_i} \sum_{k=1}^{K} I_{ij}^{(t)} I_k^{(t)} \{\log \alpha_k \\ &\qquad + \log \beta_{ij} + \log g_{k,i}^{(t)} + \log P_{ij}(\mathbf{y}^{(t)})\} . \end{aligned} \quad (16)$$

Consequently, the E-step of the EM algorithm is defined by taking the expectation of the complete-data likelihood:

$$\begin{aligned} E[l_c(\Phi; \mathcal{Y})|\mathcal{X}] = &\sum_{t=1}^{T} \sum_{i=1}^{K} \sum_{j=1}^{N_i} \sum_{k=1}^{K} h_{ij}^{(t)} h_k^{(t)} h_{k,i}^{(t)} \{\log \alpha_k \\ &+ \log \beta_{ij} + \log g_{k,i}^{(t)} + \log P_{ij}(\mathbf{y}^{(t)})\} , \end{aligned} \quad (17)$$

where $h_{ij}^{(t)}$, $h_k^{(t)}$ and $h_{k,i}^{(t)}$ are the posterior probabilities as

$$\begin{aligned} h_{ij}^{(t)} = E[I_{ij}^{(t)}|\mathcal{X}] , \quad h_k^{(t)} = E[I_k^{(t)}|\mathcal{X}] , \\ h_{k,i}^{(t)} = E[I_i^{(t)}, I_k^{(t)}|\mathcal{X}] . \end{aligned} \quad (18)$$

Computation of all the posterior probabilities is described in Appendix 2.

The M-step requires maximizing $E[l_c(\Phi; \mathcal{Y})|\mathcal{X}]$ with respect to both the expert-bank and the gate-bank parameters. By examining Eq. (17), it is apparent that the expert-bank parameters affect the $E[l_c(\Phi; \mathcal{Y})|\mathcal{X}]$ through only terms $h_{ij}^{(t)} \log P_{ij}(\mathbf{y}^{(t)})$ and $h_{ij}^{(t)} \log \beta_{ij}$, while the gate-bank parameters influence the $E[l_c(\Phi; \mathcal{Y})|\mathcal{X}]$ through only terms $h_k^{(t)} \log \alpha_k$ and $h_{k,i}^{(t)} \log g_{k,i}^{(t)}$. Thus, the M-step reduces to the following separate maximization problems:

$$\theta_{ij}^{(s+1)} = \arg \max_{\theta_{ij}} \sum_{t=1}^{T} h_{ij}^{(t)} \log P_{ij}(\mathbf{y}^{(t)}) , \quad (19)$$

$$\begin{aligned} \beta_{ij}^{(s+1)} = \arg \max_{\beta_{ij}} &\sum_{t=1}^{T} \sum_{v=1}^{N_i} h_{iv}^{(t)} \log \beta_{iv} \quad \text{s.t.} \\ &\sum_{v=1}^{N_i} \beta_{iv} = 1, \beta_{iv} \geq 0 , \end{aligned} \quad (20)$$

$$\mathbf{v}_k^{(s+1)} = \arg \max_{\mathbf{v}_k} \sum_{t=1}^{T} \sum_{i=1}^{K} h_{k,i}^{(t)} \log g_{k,i}^{(t)} , \quad (21)$$

where $\mathbf{v}_k$ is the set of all parameters of the kth gating network in the gate-bank, and

$$\begin{aligned} \alpha_k^{(s+1)} = \arg \max_{\alpha_k} &\sum_{t=1}^{T} \sum_{u=1}^{K} h_u^{(t)} \log \alpha_u \quad \text{s.t.} \\ &\sum_{u=1}^{K} \alpha_u = 1, \alpha_u \geq 0 . \end{aligned} \quad (22)$$

Problems in Eq. (19) and Eq. (21) belong to the iterative reweighted least squares (IRLS) problem. Jordan and Jacobs proposed an IRLS algorithm to solve this kind of problems.[19] However, the algorithm often suffers from instability in multiway classification. In our earlier work, the reason of instability in the IRLS algorithm was systematically investigated. Instead an improved learning algorithm has been proposed to solve this kind of IRLS problems in multiway classification[9] and, moreover, an approximation to the improved learning algorithm can be achieved for fast training[9] when a probabilistic model is subject to the generalized Bernoulli distribution.[7,9] In the case of multiway classification, therefore, problems in Eq. (19) can be solved with the improved algorithm or its approximation, while the problem in Eq. (21) can be solved only by the improved algorithm since the statistical model of gating networks is subject to the multinomial distribution, which belongs to multiway classification. As for maximization problems in Eq. (20) and Eq. (22), they can be analytically solved by

$$\beta_{ij}^{(s+1)} = \frac{1}{T} \sum_{t=1}^{T} h_{ij}^{(t)} , \quad (23)$$

and

$$\alpha_k^{(s+1)} = \frac{1}{T}\sum_{t=1}^{T} h_k^{(t)}\,. \tag{24}$$

Based on the above analysis, we summarize the EM algorithm as follows:

Algorithm: (EM Algorithm for the Generalized Mixture-of-Expert Model)

1. For each data pair $(D^{(t)}, \mathbf{y}^{(t)})$, extract K feature vectors, $\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_K^{(t)}$, from $D^{(t)}$, then compute the posterior probabilities $h_{ij}^{(t)}$, $h_k^{(t)}$, and $h_{k,i}^{(t)}$ using its current value of all parameters.
2. For each expert network, solve an IRLS problem in Eq. (19) with observations $\{(\mathbf{x}_i^{(t)}, \mathbf{y}^{(t)})\}_1^T$ and observation weights $\{h_{ij}^{(t)}\}_1^T$ by the improved learning algorithm or its approximation in terms of its probabilistic components.[9]
3. For each parameter β_{ij} in expert-banks, obtain the new estimate using Eq. (23).
4. For each gating network in the gate-bank, solve an IRLS problem in Eq. (21) with observations $\{(\mathbf{x}_u^{(t)}, h_{u,i}^{(t)})\}_1^T$ by the improved learning algorithm.[9]
5. For each parameter α_k in the gate-bank, obtain the new estimate using Eq. (24).
6. Iterate using the updated parameter values from step 1 to step 5 until a termination condition is satisfied.

3.2. *Model selection*

The pre-determined structure problem refers to that prior to training, an appropriate structure needs to be determined for a given problem. This problem has been well known in neural network community, and most of neural network models suffer from the problem. Like the standard ME architecture, this problem is also unavoidable on applying the GME model to a practical problem. To tackle the problem in the ME architecture, Jacobs *et al.* have recently proposed a pruning method based on the maximum *a posteriori* (MAP) principle for model selection.[18] Xu has proposed a so-called hard-cut EM algorithm for fast training in the ME architecture by means of his Bayesian YING-YANG learning theory.[30] The idea underlying the hard-cut EM algorithm could be extended to evaluate the usefulness of each expert network for a given problem during training. Motivated by their work, we propose an alternative pruning method for the GME architecture.

For a given classification problem, we assume that the initial GME structure is always of a complicated topology. The idea underlying the proposed pruning method is to select an appropriate model by combination of the MAP and the cross-validation principles. According to the posterior probabilities, we can transfer the soft-competition scheme into a winner-take-all mechanism in our architecture. The winner is defined as the expert network with the maximal posterior probability for a sample $D^{(t)}$ in a training set with T samples. In other words, a label with binary value can be assigned to each expert network in terms of the sample $D^{(t)}$ to indicate whether it is a winner or not. For expert network (i, j), the value of the label, $z_{ij}^{(t)}$, is defined as

$$z_{ij}^{(t)} = \begin{cases} 1 & \text{if } h_{ij}^{(t)} = \arg\max\limits_{uv} h_{uv}^{(t)} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where $u = 1, \ldots, K$, $v = 1, \ldots, N_u$ (K is the number of expert-banks and N_u is the number of expert networks in expert-bank u), and $h_{uv}^{(t)}$ is the posterior probability as defined in Eq. (18). Moreover, a winner-index WI_{ij} based on the labels z_{ij} for expert network (i, j) is defined as

$$WI_{ij} = \frac{1}{T}\sum_{t=1}^{T} z_{ij}^{(t)}\,. \tag{26}$$

Based on the winner-index, we present a two-stage learning procedure with the pruning mechanism for parameter estimation and model selection in the GME architecture. The learning procedure is described as follows: (a) For a given problem, a GME structure starting with the complicated topology is first trained on a training set by the EM algorithm described in Sec. 3.1. (b) Once the training is finished, the winner-index for each expert network is calculated on a cross-validation set. Any expert network (i^*, j^*) will be pruned if $WI_{i^*j^*} < \varepsilon$, where ε is a pre-specified threshold. (c) The parameter values achieved in step a remain as initialization, and the resulting GME structure achieved in Step b is re-trained on the training set used in step a by the EM algorithm.

In contrast to the pruning method reported in Ref. 18, we use a two-stage learning procedure for

learning and pruning a GME structure based on a cross-validation data set, which may lead to better generalization. We point out that this two-stage learning procedure might be used in the alternative mixture-of-expert architecture developed in Ref. 3 as well.

4. Simulations

The generalized mixture-of-expert architecture has been applied to a real world problem called text-dependent speaker identification. All the simulations were done in a SUN SPARC II workstation. In this section, we first describe the speaker identification problem, and then present simulation results. Finally, we report comparative results produced by related methods.

4.1. *Text-dependent speaker identification*

Speaker identification is to classify an unlabeled voice token as belonging to one of reference speakers. In particular, text-dependent speaker identification refers to that the same text is used in both training and testing phases. The problem is a hard classification task because a person's voice always changes over time. The main outcome of many studies on speaker's features indicates that the speech spectrum reflects the anatomical structure of a person's vocal tract and nasal cavities and so contains information about hopefully unique physical attributes.[10,13,14,23,26,27] However, there is less agreement on which parameterization of the speech spectrum to use as features for speaker identification. Many kinds of spectral features have been reported to be useful to speaker identification. Therefore, speaker identification becomes a typical problem of pattern classification based on different feature sets. In earlier studies, two or more kinds of different feature sets were combined together to form a composite feature set so that different feature sets can be used simultaneously.[15,22] However, the performance of such a system based on a composite feature set is not significantly improved. On the other hand, methods of combining multiple classifiers trained on different feature sets were recently applied to speaker identification, which led to satisfactory results.[4,5] However, a large amount of data were demanded for training not only multiple classifiers but also a combination scheme.

In the simulations, we chose isolated digits as the fixed text. The method has been extensively used in text-dependent speaker identification.[3,4,6–8,10,28] The acoustic database consisted of ten isolated digits from '0' to '9' uttered in Mandarin. All utterances were recorded in three different sessions and ten male speakers were registered in the database. For each digit, 100 utterances (10 utterances/speaker) were recorded in each session. We divided all data into three data sets in terms of recording sessions for different use. The technical details of preprocessing are briefly as follows: (a) 16-bit A/D-converter with 11.025 kHz sampling rate, (b) processing the data with a pre-emphasis filter $H(z) = 1 - 0.95z^{-1}$ and (c) 25.6 msec Hamming window with 12.8 msec overlapping for blocking an utterance into several feature frames in the short-time spectral analysis. Thus, an utterance of digit was blocked as a sequence of frames. In the simulations, we adopted four common speech spectral features for speaker identification, i.e., 19-order delta-cepstrum (DEL-CEPS), 19-order LPC based cepstrum (LPC-CEPS), 15-order LPC coefficients (LPC-COEF), and 19-order Mel-scale cepstrum (MEL-CEPS). After preprocessing, the four different feature vectors were independently extracted from each frame.[25]

4.2. *Results of the GME architecture*

In the simulations, ten GME classifiers were employed where each GME classifier was used to handle one of ten digits in terms of a feature set chosen. The initial structure of these GME classifiers consisted of four expert-banks (five expert networks in each expert-bank) and a gate-bank with four gating networks due to four different feature sets used simultaneously for identification (see Fig. 2). In particular, four kinds of feature vectors, i.e., DEL-CEPS, LPC-CEPS, LPC-COEF, and MEL-CEPS, are input to expert-banks 1–4, respectively. Since the current identification problem is a special multiway classification in which each component of output is binary and there is only a single non-zero component, the generalized Bernoulli distribution[7,9] can be used as the probabilistic model of expert networks. We used the improved learning algorithm to train all the gating networks and its approximation to train

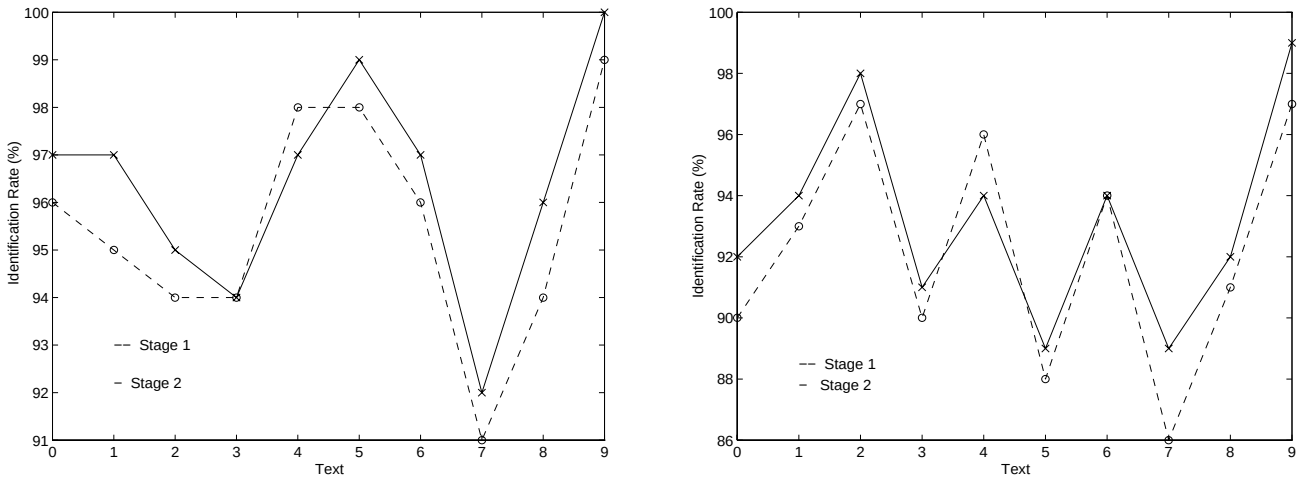


Fig. 3. Identification rates of ten GME classifiers corresponding to ten digits in terms of two learning stages in the digit-based test. (a) Results of Test-1. (b) Results of Test-2.

all the expert networks in the M-step of the EM algorithm.[9] All utterances recorded in the first session were used during training; i.e., the training set consisted of 60 utterances (6 utterances/speaker) for each digit, and the cross-validation set was composed of the remaining 40 utterances. All the utterances recorded in two additional sessions were used for test, and tests on the two data sets recorded in the second and third sessions were called Test-1 and Test-2, accordingly.

In the simulations, we adopted two testing methods: digit-based test and sequence-based test. In the digit-based test, the utterance of one single digit was used for identifying an unknown speaker. Since an utterance was divided into several frames in the short-time spectral analysis, the mean of results produced by all the feature vectors belonging to one utterance was used as the final identification result of the utterance. In contrast, a sequence of five isolated digits (it may be viewed as a password) were used in the sequence-based test. For each digit, an identifying result can be achieved based on the digit-based method. After achieving all five individual results, the system polled a vote with the majority principle that an unknown speaker can be identified only when at least three of the five GME classifiers produced the same identification results. Otherwise, the system would reject the unknown speaker.

Figure 3 shows identification rates produced by ten GME classifiers in two learning stages of our learning algorithm presented in Sec. 3.2 by the digit-based test. The CPU time of training GME classifiers in two learning stages is illustrated in Fig. 4. It is evident from the simulations that the generalization performance of resulting structures is improved in general using the two-stage learning procedure, and the training in the second stage

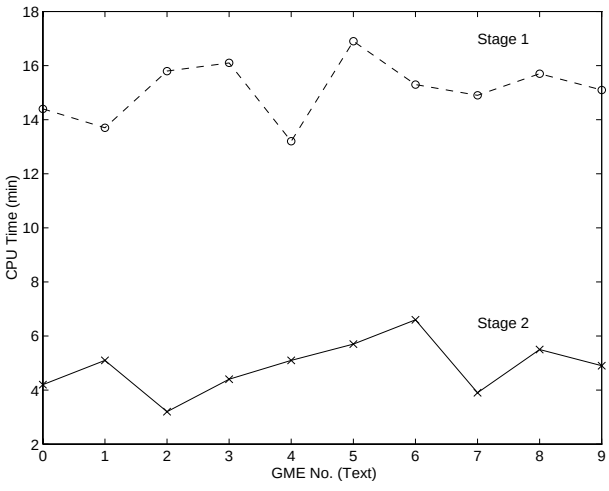
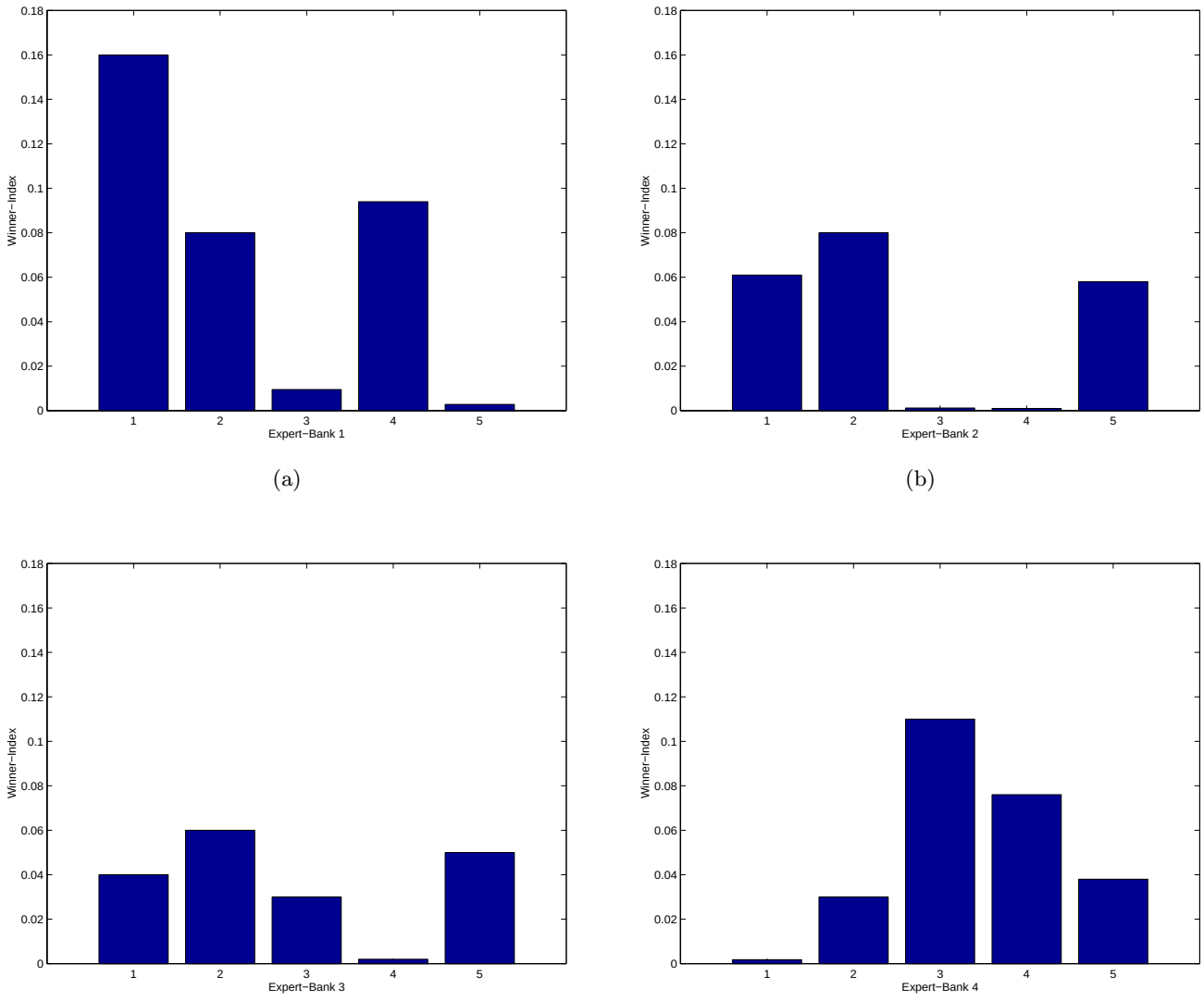


Fig. 4. CPU time of training ten GME classifiers corresponding to ten digits taken in two learning stages.



(a) (b) (c) (d)

Fig. 5. The winner-index values of 20 expert networks located in four expert-banks of the GME classifier corresponding to digit '9'. Initially, there are five expert networks in each expert-bank. (a) Results of expert-bank 1. (b) Results of expert-bank 2. (c) Results of expert-bank 3. (d) Results of expert-bank 4.

is considerably faster than that in the first stage because the achieved parameters in the first stage were used to initialize the pruned structure in the second stage. To demonstrate the pruning process, Fig. 5 illustrates the winner-index values of all expert networks in the GME classifier corresponding to digit '9' for instance. In our simulations, we chose the threshold ε as 0.005. Here the value of ε implies that an expert network will be pruned from the initial structure when the number that the expert network becomes winners is less than five for test by a cross-validation set of 1000 samples. As a result, the remaining numbers of expert networks in expert-banks 1–4 are 4, 3, 4, and 4, respectively, after pruning. Due to limited space, we report only the resulting structures of all GME classifiers after pruning in Table 1 instead of their winner-index values.

Furthermore, we used the sequence-based method to evaluate the performance. As shown in Table 2, the testing results show that the speaker

Table 1. Numbers of expert networks in ten GME classifiers after pruning.

GME Classifiers	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
Expert-Bank 1	5	3	5	4	4	4	4	4	5	4
Expert-Bank 2	3	4	3	3	3	4	4	2	3	3
Expert-Bank 3	4	3	4	2	3	4	2	3	3	4
Expert-Bank 4	3	4	5	5	4	4	4	4	4	4

Table 2. Performance of the speaker identification system based on the GME architecture by the sequence-based test.

Test No.	500	1000	2000	3000	4000	5000
Identification No.	500	1000	1999	2998	3995	4994
Substitution No.	0	0	0	1	2	2
Rejection No.	0	0	1	1	3	4

identification system based on the GME architecture can be practically used with acceptable performance.

4.3. *Comparative results*

For comparison, we have already conducted some simulations by applying related methods to the same problem. The training and cross-validation sets were the same as described above. It is obvious that the performance of the sequence-based test highly depends upon that of the digit-based test. Therefore, we merely report simulation results of the digit-based test in the sequel.

First of all, we used the standard ME architecture[17] as classifiers to deal with the same problem. Simulations were two-fold; four individual different feature sets were independently used and a 72-dimensional composite feature set formed by combination of the four different feature sets together was used to train ME classifiers. Structures of ten ME classifiers were determined by the cross-validation method. All structures of ME classifiers ranging from 12 to 16 experts were investigated and the 'optimal' structures used in the simulations are shown in Table 3, where for the same digit the use of different feature sets as input may result in different structures of ME classifiers. Similarly, the improved learning algorithm and its approximation were used in the M-step of the EM algorithm to train the gating network and expert networks. Figure 6 shows testing results in Test-1 and Test-2, and Fig. 7 illustrates CPU time of training the ME classifiers. Due to limited space, we only show the mean identification rates and the mean training time of ME classifiers trained on individual feature sets in Figs. 6 and 7. For comparison, the identification rates and total training time of the two-stage learning in those GME classifiers are also shown in Figs. 6 and 7. It is evident from the simulation results that the GME architecture outperforms the ME architecture in terms

Table 3. Structures of ME classifiers used in the comparative experiments: the number of expert networks in each ME classifier.

ME Classifiers	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
DEL-CEPS	12	13	12	12	13	14	14	13	14	16
LPC-CEPS	13	13	13	13	14	14	14	13	15	15
LPC-COEF	13	12	12	13	12	12	13	13	13	14
MEL-CEPS	14	14	15	13	14	14	16	14	14	15
Composite Feature	16	15	16	14	15	15	16	16	15	16

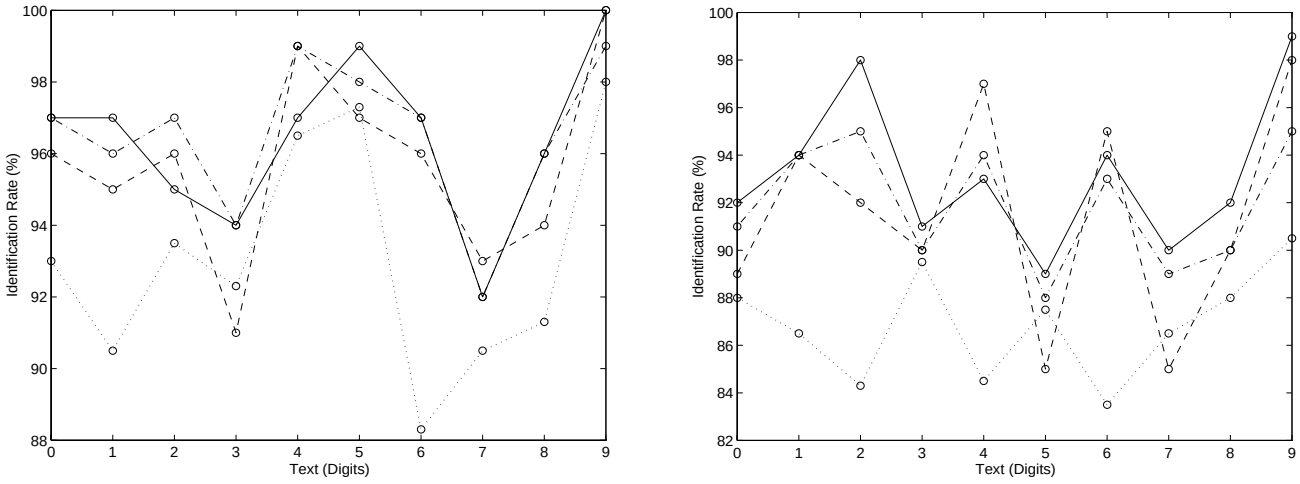


Fig. 6. Comparative results: identification rates of the GME architecture (solid line), ME classifiers trained on either individual feature sets (dotted line), ME classifiers trained on a composite feature set (dashed line), and the Bayesian reasoning combination of ME classifiers trained on different feature sets (dash-dot line). (a) Results of Test-1. (b) Results of Test-2.

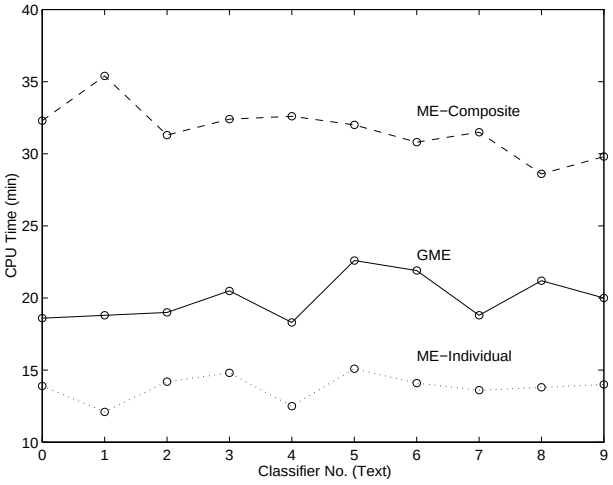


Fig. 7. Comparative results: CPU time of training the GME classifiers and ME classifiers trained on either individual feature sets or a composite feature set.

of either the use of individual feature sets or the composite feature set. In particular, the GME classifiers yield significantly fast training in comparison with ME classifiers on the composite feature set though the training of the GME classifiers is slightly slower than that of ME classifiers on individual feature sets due to the two-stage learning.

We also conducted an experiment of combining four ME classifiers trained on different feature sets for the same problem. The so-called Bayesian reasoning combination method[31] was adopted as the combination scheme. This method is viewed as a typical application of the stacked generalization principle[29] to improve the performance of multiple classifiers. The recent investigation showed that the Bayesian reasoning combination method yields the best performance on a benchmark hand-written OCR problem in contrast to other typical combination methods.[24,31] In this combination method, the confusion matrix[31] was estimated based on the cross-validation set for combination. The identification rates of the combination method are also shown in Fig. 6 for comparison. According to simulation results, we found that in the terms of the mean identification rates on ten digits, the combination method slightly outperforms the GME architecture in Test-1, but its performance is worse than the GME architecture in Test-2. A possible reason of this instable outcome is due to insufficient data used to estimate the confusion matrix according to our previous work on the combination of multiple classifiers trained on different feature sets.[5]

In summary, comparative results reported above indicate that our model yields better performance in text-dependent speaker identification. In particular, our model leads to robust performance in the case of less data available for training.

5. Conclusions

We have described a novel modular neural network architecture to implement the soft competition scheme on different feature sets. For learning, an EM algorithm is developed for adjusting the parameters in our architecture and, moreover, a model selection method is proposed to obtain an appropriate structure for a specific problem. The EM algorithm and the model selection method constitute a two-stage learning procedure so that an initial structure can be pruned to produce better generalization performance. Simulation results that the proposed method yields satisfactory performance and fast training in contrast to related methods.

Acknowledgments

We wish to thank Professor L. Xu for his comments that led to improvements in this manuscript. This work was supported in part by National Science Foundation in China under the grant 69635020 and National 973 Key Fundamental Research Project in China under the grant G1999032708.

Appendix 1. Generalized Hierarchical Mixtures of Experts

In this appendix, we extend the GME architecture to a hierarchical structure, which can be viewed as a generalized hierarchical mixture-of-expert (GHME) model for pattern classification based on different feature sets. To simplify the presentation, we restrict ourselves to a two-level hierarchy throughout the appendix. All of the algorithms described in the appendix, however, can generalize readily to hierarchies of arbitrary depth. One may be referred to Ref. 20 for a recursive formalism that handles arbitrary hierarchies.

A two-level GHME architecture consisting of N GME modules (cf. Fig. 2) is a tree in which the gate-banks sit at the nonterminals of the tree and expert-banks sit the leaves of the tree. These gate-banks receive the vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_K$, as input and produce scalar outputs assuming that K different feature sets can be extracted. K expert-banks are used to handle different feature sets in each GME module, respectively. The ith expert-bank in the mth GME module consists of $N_{i|m}$ expert networks which receive the same input vector, $\mathbf{x}_i$, while different expert-banks receive the different feature vectors. The output vector of an expert-bank is the convex weighted sum of output vectors produced by expert networks in the expert-bank. Output vectors of all expert-banks proceed up the tree, being blended by the gate-bank outputs to produce the final output of the GHME architecture.

The jth expert network associated with the ith expert-bank in the mth GME module produces the output, $\mathbf{o}_{ij|m}$, as

$$\mathbf{o}_{ij|m} = f(W_{ij|m}\mathbf{x}_i)\,, \tag{27}$$

where $W_{ij|m}$ is a weight matrix. The output of the ith expert-bank in the mth GME module, $\mathbf{o}_{i|m}$, is

$$\mathbf{o}_{i|m} = \sum_{j=1}^{N_{i|m}} \beta_{ij|m}\mathbf{o}_{ij|m}\,, \tag{28}$$

where $\beta_{ij|m}$ are linear coefficients for combining outputs produced by expert networks in the ith expert-bank on the conditions: $\sum_{j=1}^{N_i} \beta_{ij|m} = 1$ and $\beta_{ij|m} \geq 0$.

The mth output of the nth gating network in the top-level gate-bank, $g_{n,m}$, is

$$g_{n,m} = \frac{e^{\xi_{n,m}}}{\sum_{u=1}^{N} e^{\xi_{n,u}}}, \tag{29}$$

where $\xi_{n,m} = \mathbf{v}_{n,m}^T\mathbf{x}_n$ and $\mathbf{v}_{n,m}$ is a weight vector. Furthermore, the mth output of the top-level gate-bank λ_m is the convex weighted sum of $g_{n,m}$:

$$\lambda_m = \sum_{n=1}^{K} \alpha_n g_{n,m}\,, \tag{30}$$

where α_n are linear coefficients for combining outputs produced by gating networks in the top-level gate-bank on the conditions: $\sum_{k=1}^{K} \alpha_n = 1$ and $\alpha_n \geq 0$. Similarly, the ith output of the kth gating network in the lower-level gate-bank in the mth

GME module, $g_{k,i|m}$, is

$$g_{k,i|m} = \frac{e^{\xi_{k,i|m}}}{\sum_{u=1}^{K} e^{\xi_{k,u|m}}}, \tag{31}$$

where $\xi_{k,i|m} = \mathbf{v}_{k,i|m}^T \mathbf{x}_k$ and $\mathbf{v}_{k,i|m}$ is a weight vector. Furthermore, the ith output of the lower-level gate-bank in the mth GME module, $\lambda_{i|m}$, is

$$\lambda_{i|m} = \sum_{k=1}^{K} \alpha_{k|m} g_{k,i|m}, \tag{32}$$

where $\alpha_{k|m}$ are linear coefficients for combining outputs produced by gating networks in the gate-bank on the condition that $\sum_{k=1}^{K} \alpha_{k|m} = 1$ and $\alpha_{k|m} \geq 0$. Therefore, the total output of the GHME architecture, $\mathbf{o}$, is

$$\mathbf{o} = \sum_{m=1}^{N} \lambda_m \sum_{i=1}^{K} \lambda_{i|m} \mathbf{o}_{i|m}. \tag{33}$$

Let $P(\mathbf{y}|\mathbf{x}_i, \theta_{i|m})$ denote the probability model of the ith expert-bank in the mth module in the GHME architecture, where $\theta_{i|m}$ denotes the set of parameters in the model. Let $P(\mathbf{y}|\mathbf{x}_i, \theta_{ij|m})$ denote the probability model of the jth expert network associated with the ith expert-bank in the mth module in the GHME architecture, where $\theta_{ij|m}$ is the set of parameters in the model. The generalized finite mixture model of the GHME architecture is specified by

$$\begin{aligned} P(\mathbf{y}|D, \Phi) &= \sum_{m=1}^{N} \lambda_m \sum_{i=1}^{K} \lambda_{i|m} P(\mathbf{y}|\mathbf{x}_i, \theta_{i|m}) \\ &= \sum_{m=1}^{N} \sum_{n=1}^{K} \alpha_n g_{n,m}(\mathbf{x}_n, \mathbf{v}_{n,m}) \\ &\quad \times \sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m} g_{k,i|m}(\mathbf{x}_k, \mathbf{v}_{k,i|m}) \\ &\quad \times \beta_{ij|m} P(\mathbf{y}|\mathbf{x}_i, \theta_{ij|m}), \end{aligned} \tag{34}$$

where Φ includes all the expert-bank parameters including $\theta_{ij|m}$ and $\beta_{ij|m}$ and the gate-bank parameters including $\mathbf{v}_{n,m}$, α_n, $\mathbf{v}_{k,i|m}$ and $\alpha_{k|m}$. In the remainder of this appendix we drop the explicit reference to the input and the parameters to simplify the notation. As a result, the probability model in Eq. (34) is rewritten as

$$\begin{aligned} P(\mathbf{y}|D, \Phi) = \sum_{m=1}^{N} \sum_{n=1}^{K} \alpha_n g_{n,m} \sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \\ \times \alpha_{k|m} g_{k,i|m} \beta_{ij} P_{ij|m}(\mathbf{y}). \end{aligned} \tag{35}$$

For the GHME architecture, we develop an EM algorithm for adjusting the parameters in this architecture. Given the current estimate $\Phi^{(s)}$, each epoch consists of the E-step and the M-step as follows:

(1) **E-step**

For each pair $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$, the posterior probabilities $h_m^{(t)}$, $h_n^{(t)}$, $h_{k|m}^{(t)}$, $h_{i|m}^{(t)}$ and $h_{k,i|m}^{(t)}$ are computed as

$$h_m^{(t)} = \frac{g_{n,m}^{(t)} \sum_{n=1}^{K} \alpha_n^{(s)} \sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}{\sum_{m=1}^{N} \sum_{n=1}^{K} \alpha_n^{(s)} g_{n,m}^{(t)} \sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}, \tag{36}$$

$$h_n^{(t)} = \frac{\alpha_n^{(s)} \sum_{m=1}^{N} g_{n,m}^{(t)} \sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}{\sum_{m=1}^{N} \sum_{n=1}^{K} \alpha_n^{(s)} g_{n,m}^{(t)} \sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}, \tag{37}$$

$$h_{k|m}^{(t)} = \frac{\alpha_{k|m}^{(s)} \sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}{\sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}, \tag{38}$$

$$h_{ij|m}^{(t)} = \frac{\sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}{\sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}, \tag{39}$$

and

$$h_{k,i|m}^{(t)} = \frac{\alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}{\sum_{i=1}^{K} \sum_{j=1}^{N_{i|m}} \sum_{k=1}^{K} \alpha_{k|m}^{(s)} g_{k,i|m}^{(t)} \beta_{ij|m}^{(s)} P_{ij|m}(\mathbf{y}^{(t)})}. \tag{40}$$

(2) **M-step**

A new estimate $\Phi^{(s+1)}$ is found through solving the following maximization problems:

$$\theta_{ij|m}^{(s+1)} = \arg\max_{\theta_{ij|m}} \sum_{t=1}^{T} h_m^{(t)} h_{ij|m}^{(t)} \log P_{ij|m}(\mathbf{y}^{(t)}), \tag{41}$$

$$\alpha_n^{(s+1)} = \arg\max_{\alpha_n} \sum_{t=1}^{T} \sum_{u=1}^{K} h_u^{(t)} \log \alpha_u, \quad \text{s.t.} \sum_{u=1}^{K} \alpha_u = 1,\ \alpha_u \geq 0, \tag{42}$$

$$\mathbf{v}_{n,m}^{(s+1)} = \arg\max_{\mathbf{v}_{n,m}} \sum_{t=1}^{T} \sum_{z=1}^{N} \sum_{w=1}^{K} h_{w,z}^{(t)} \log g_{w,z}^{(t)} \tag{43}$$

$$\alpha_{k|m}^{(s+1)} = \arg\max_{\alpha_{k|m}} \sum_{t=1}^{T} \sum_{u=1}^{K} h_{u|m}^{(t)} \log \alpha_{u|m}, \quad \text{s.t.} \sum_{u=1}^{K} \alpha_{u|m} = 1,\ \alpha_{u|m} \geq 0, \tag{44}$$

$$\mathbf{v}_{k|m}^{(s+1)} = \arg\max_{\mathbf{v}_{k|m}} \sum_{t=1}^{T} \sum_{z=1}^{N} h_z^{(t)} \sum_{u=1}^{K} \sum_{w=1}^{K} h_{w,u|z}^{(t)} \log g_{w,u|z}^{(t)}, \tag{45}$$

where $\mathbf{v}_{k|m}$ is the set of all the parameters of the kth gating network in a lower-level gate-bank, and

$$\beta_{ij|m}^{(s+1)} = \arg\max_{\beta_{ij|m}} \sum_{t=1}^{T} \sum_{v=1}^{N_{i|m}} h_{iv|m}^{(t)} \log \beta_{iv|m}, \quad \text{s.t.} \sum_{v=1}^{N_{i|m}} \beta_{iv|m} = 1,\ \alpha_{iv|m} \geq 0. \tag{46}$$

Problems in Eq. (41), Eq. (43) and Eq. (45) belong to the IRLS problem which can be solved using the improved learning algorithm or its approximation.[9] The maximization problems in Eq. (42), Eq. (44) and Eq. (46) can be analytically solved, respectively, as

$$\alpha_n^{(s+1)} = \frac{1}{T} \sum_{t=1}^{T} h_n^{(t)}, \tag{47}$$

$$\alpha_{k|m}^{(s+1)} = \frac{1}{T} \sum_{t=1}^{T} h_{k|m}^{(t)}, \tag{48}$$

and

$$\beta_{ij|m}^{(s+1)} = \frac{1}{T}\sum_{t=1}^{T} h_{ij|m}^{(t)}\,. \tag{49}$$

Appendix 2. Posterior Probabilities in EM Algorithm

In this appendix, we derive the posterior probabilities used in the E-step of the EM algorithm for the GME architecture.

$E[I_{ij}^{(t)}|\mathcal{X}]$ can be computed using the Bayesian rule as

$$\begin{aligned} E[I_{ij}^{(t)}|\mathcal{X}] &= P(I_{ij}^{(t)} = 1|\mathbf{y}^{(t)}, D^{(t)}, \Phi^{(s)}) \\ &= P(I_i^{(t)} = 1, I_{j|i}^{(t)} = 1|\mathbf{y}^{(t)}, D^{(t)}, \Phi^{(s)}) \\ &= \frac{P(\mathbf{y}^{(t)}|I_i^{(t)} = 1, I_{j|i}^{(t)} = 1, D^{(t)}, \Phi^{(s)})P(I_i^{(t)} = 1, I_{j|i}^{(t)} = 1|D^{(t)}, \Phi^{(s)})}{P(\mathbf{y}^{(t)}|D^{(t)}, \Phi^{(s)})}\,. \end{aligned} \tag{50}$$

Furthermore, $P(I_i^{(t)} = 1, I_{j|i}^{(t)} = 1|D^{(t)}, \Phi^{(s)})$ can be computed by the joint and total probability rules as

$$P(I_i^{(t)} = 1, I_{j|i}^{(t)} = 1|D^{(t)}, \Phi^{(s)}) = P(I_{j|i}^{(t)} = 1|I_i^{(t)} = 1, D^{(t)}, \Phi^{(s)})P(I_i^{(t)} = 1|D^{(t)}, \Phi^{(s)})\,, \tag{51}$$

and

$$P(I_i^{(t)} = 1|D^{(t)}, \Phi^{(s)}) = \sum_{k=1}^{K} P(I_i^{(t)} = 1|I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)})P(I_k^{(t)} = 1|D^{(t)}, \Phi^{(s)})\,. \tag{52}$$

According to the probabilistic model as defined in Eq. (9), we have

$$P(I_k^{(t)} = 1|I_i^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = g_{k,i}^{(t)}\,, \tag{53}$$

$$P(I_k^{(t)} = 1|D^{(t)}, \Phi^{(s)}) = \alpha_k^{(s)}\,, \tag{54}$$

$$P(I_{j|i}^{(t)} = 1|I_i^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = \beta_{ij}^{(s)}\,, \tag{55}$$

and

$$P(\mathbf{y}^{(t)}|I_{ij}^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = P(\mathbf{y}^{(t)}|I_i^{(t)} = 1, I_{j|i}^{(t)} = 1, D^{(t)}, \Phi^{(s)}) = P_{ij}(\mathbf{y}^{(t)})\,. \tag{56}$$

Therefore, inserting Eqs. (51) and (52) into Eq. (50) yields

$$E[I_{ij}^{(t)}|\mathcal{X}] = \frac{\sum_{k=1}^{K} \alpha_k^{(s)} g_{k,i}^{(t)} P_{ij}\beta_{ij}^{(s)}(\mathbf{y}^{(t)})}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K} \alpha_k^{(s)} g_{k,i}^{(t)}\beta_{ij}^{(s)} P_{ij}(\mathbf{y}^{(t)})}\,. \tag{57}$$

$E[I_k^{(t)}|\mathcal{X}]$ is computed using the Bayesian rule as

$$\begin{aligned} E[I_k^{(t)}|\mathcal{X}] &= P(I_k^{(t)} = 1|\mathbf{y}^{(t)}, D^{(t)}, \Phi^{(s)}) \\ &= \frac{P(\mathbf{y}^{(t)}|I_k^{(t)} = 1, D^{(t)}, \Phi^{(s)})P(I_k^{(t)} = 1|D^{(t)}, \Phi^{(s)})}{P(\mathbf{y}^{(t)}|D^{(t)}, \Phi^{(s)})}\,. \end{aligned} \tag{58}$$

Furthermore, $P(\mathbf{y}^{(t)}|I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})$ can be computed by the total probability rule as

$$P(\mathbf{y}^{(t)}|I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)}) = \sum_{i=1}^{K}\sum_{j=1}^{N_i} P(\mathbf{y}^{(t)}|I_k^{(t)}=1,\, I_{ij}^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})P(I_{ij}^{(t)}=1|I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})$$

$$= \sum_{i=1}^{K}\sum_{j=1}^{N_i} P(\mathbf{y}^{(t)}|I_{ij}^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})P(I_{ij}^{(t)}=1|I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})\,. \tag{59}$$

Note that the indicator variable $I_k^{(t)}=1$ can be ignored from $P(\mathbf{y}^{(t)}|I_k^{(t)}=1,\, I_{ij}^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})$ in Eq. (59) since it is independent of the probabilistic model based on the fact that $\mathbf{y}^{(t)}$ is generated from the jth expert network in the ith expert-bank regardless of any gating network. According to Eqs. (53)–(56), inserting Eq. (59) into Eq. (58) yields

$$E[I_k^{(t)}|\mathcal{X}] = \frac{\alpha_k^{(s)}\sum_{i=1}^{K}\sum_{j=1}^{N_i} g_{k,i}^{(t)}\beta_{ij}^{(s)}P_{ij}(\mathbf{y}^{(t)})}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k^{(s)}g_{k,i}^{(t)}\beta_{ij}^{(s)}P_{ij}(\mathbf{y}^{(t)})}\,. \tag{60}$$

$E[I_i^{(t)}=1,\, I_k^{(t)}|\mathcal{X}]$ can be computed using the Bayesian rule as

$$E[I_i^{(t)},\, I_k^{(t)}|\mathcal{X}] = P(I_i^{(t)}=1,\, I_k^{(t)}=1|\mathbf{y}^{(t)},\, D^{(t)},\, \Phi^{(s)})$$

$$= \frac{P(\mathbf{y}^{(t)}|I_i^{(t)}=1,\, I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})P(I_i^{(t)}=1,\, I_k^{(t)}=1|D^{(t)},\, \Phi^{(s)})}{P(\mathbf{y}^{(t)}|D^{(t)},\, \Phi^{(s)})}\,. \tag{61}$$

Similarly, $P(I_i^{(t)}=1,\, I_k^{(t)}=1|D^{(t)},\, \Phi^{(s)})$ can be computed as

$$P(I_i^{(t)}=1,\, I_k^{(t)}=1|D^{(t)},\, \Phi^{(s)}) = P(I_i^{(t)}=1|I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})P(I_k^{(t)}=1|D^{(t)},\, \Phi^{(s)})\,, \tag{62}$$

and $P(\mathbf{y}^{(t)}|I_i^{(t)}=1,\, I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})$ can be computed using the total probability rule as

$$P(\mathbf{y}^{(t)}|I_i^{(t)}=1,\, I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})$$

$$= \sum_{j=1}^{N_i} P(\mathbf{y}^{(t)}|I_i^{(t)}=1,\, I_{j|i}^{(t)}=1,\, I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})P(I_{j|i}^{(t)}=1|I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})$$

$$= \sum_{j=1}^{N_i} P(\mathbf{y}^{(t)}|I_i^{(t)}=1,\, I_{j|i}^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})P(I_{j|i}^{(t)}=1|I_k^{(t)}=1,\, D^{(t)},\, \Phi^{(s)})\,. \tag{63}$$

According to Eqs. (53)–(56), inserting Eqs. (62) and (63) into Eq. (61) yields

$$E[I_i^{(t)},\, I_k^{(t)}|\mathcal{X}] = \frac{\alpha_k^{(s)}g_{k,i}^{(t)}\sum_{j=1}^{N_i}\beta_{ij}^{(s)}P_{ij}(\mathbf{y}^{(t)})}{\sum_{i=1}^{K}\sum_{j=1}^{N_i}\sum_{k=1}^{K}\alpha_k^{(s)}g_{k,i}^{(t)}\beta_{ij}^{(s)}P_{ij}(\mathbf{y}^{(t)})}\,. \tag{64}$$

References

1. R. Battiti and A. M. Colla 1994, "Democracy in neural nets: Voting schemes for classification," *Neural Networks* **7**(4), 691–708.
2. A. Blum 1997, "Selection of relevant features and examples in machine learning," *Artificial Intelligence* **97**(2), 245–272.
3. K. Chen 1998, "A connectionist method for pattern classification with diverse features," *Pattern Recognition Letters* **19**(7), 545–558.
4. K. Chen and H. Chi 1998, "A method of combining multiple probabilistic classifiers through soft competition on different feature sets," *Neurocomputing* **20**(1–3), 227–252.
5. K. Chen, L. Wang and H. Chi 1997, "Methods of combining multiple classifiers with different features and their applications to text-independent speaker recognition," *Int. J. Pattern Recognition and Artificial Intell.* **11**(3), 417–445.
6. K. Chen, D. Xie and H. Chi 1996, "A modified HMEs for text-dependent speaker identification," *IEEE Trans. Neural Networks* **7**(5), 1309–1313.
7. K. Chen, D. Xie and H. Chi 1996, "Speaker identification using time-delay HMEs," *Int. J. Neural Systems* **7**(1), 29–43.
8. K. Chen, D. Xie and H. Chi 1996, "Text-dependent speaker identification based on input/output HMMs: An empirical study," *Neural Processing Letters* **3**(2), 81–89.
9. K. Chen, L. Xu and H. Chi 1999, "Improved learning algorithms for mixtures of experts in multiclass classification," *Neural Networks* **12**(9), 1229–1252.
10. G. Doddington 1986, "Speaker recognition — identifying people by their voice," *Proc. IEEE* **73**(11), 1651–1664.
11. R. Duda and P. Hart 1973, *Pattern Classification and Scene Analysis* (John Wiley & Sons, New York).
12. F. Fogelman-Soulie, E. Viennet and B. Lamy 1993, "Multi-modular neural network architectures: Applications in optical character and human face recognition," *Int. J. Pattern Recognition and Artificial Intell.* **7**(4), 521–555.
13. S. Furui 1981, "Comparison of speaker recognition methods using statistical features and dynamic features," *IEEE Trans. Acoust., Speech and Signal Processing* **29**(3), 197–200.
14. S. Furui 1986, "Research on individual features in speech waves and automatic speaker recognition techniques," *Speech Communication* **5**(2), 183–197.
15. S. Furui 1994, "An overview of speaker recognition technology," in *Proceedings of ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pp. 1–9.
16. R. A. Jacobs 1995, "Methods for combining experts' probability assessments," *Neural Computation* **7**(5), 867–888.
17. R. A. Jacobs, M. I. Jordan, S. Nowlan and G. Hinton 1991, "Adaptive mixture of local experts," *Neural Computation* **3**, 79–87.
18. R. A. Jacobs, F. Peng and M. A. Tanner 1997, "A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures," *Neural Networks* **10**(2), 231–241.
19. M. I. Jordan and R. A. Jacobs 1994, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation* **6** 181–214.
20. M. I. Jordan and L. Xu 1995, "Convergence results for the EM approach to mixtures of experts," *Neural Networks* **8**(9), 1409–1431.
21. R. Meir 1994, Bias, variance, and the combination of estimators. Tech. Rep. 922, Department of Electrical Engineering, Technion, Haifa, Israel.
22. J. P. Openshaw, Z. P. Sun and J. S. Mason 1993, "A comparison of composite features under degraded speech in speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, II371–II374.
23. D. O'Shaughnessy 1986, "Speaker recognition," *IEEE ASSP Magazine* **3**(4), 4–17.
24. M. P. Perrone 1993, Improving regression estimation: averaging methods of variance reduction with extensions to general convex measure optimization. Ph.D. thesis, Department of Physics, Brown University.
25. L. R. Rabiner and B. H. Juang 1993, *Fundamentals of Speech Recognition* (Englewood Cliffs, Prentice-Hall, New Jersey).
26. D. A. Reynolds 1992, A Gaussian mixture modeling approach to text-independent speaker identification. Ph.D. Thesis, Department of Electrical Engineering, Georgia Institute of Technology.
27. M. R. Sambur 1975, "Selection of acoustic features for speaker identification," *IEEE Trans. Acoust., Speech and Signal Processing* **23**, 176–182.
28. F. K. Soong and A. E. Rosenberg 1988, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Trans. Acoust., Speech and Signal Processing* **36**(6), 871–879.
29. D. H. Wolpert 1992, "Stacked generalization," *Neural Networks* **5**, 241–259.
30. L. Xu 1996, "Bayesian-Kullback YING-YANG machines for supervised learning," in *Proceedings of World Congress of Neural Networks*, 193–200.
31. L. Xu, A. Krzyzak and C. Y. Suen 1992, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. on System, Man and Cybern.* **23**(3), 418–435.
32. Z. Zhang, I. Harmann, J. Guo and R. Suchenwirth 1989, "A recognition method of printed Chinese character by feature combination," *International Journal of Research and Engineering – Postal Applications* **1**, 77–82.