Estimation of Elliptical Basis Function Parameters by the EM Algorithm with Application to Speaker Verification

Man-Wai Mak, *Member, IEEE*, and Sun-Yuan Kung, *Fellow, IEEE*

Abstract—**This paper proposes to incorporate full covariance matrices into the radial basis function (RBF) networks and to use the expectation-maximization (EM) algorithm to estimate the basis function parameters. The resulting networks, referred to as elliptical basis function (EBF) networks, are evaluated through a series of text-independent speaker verification experiments involving 258 speakers from a phonetically balanced, continuous speech corpus (TIMIT). We propose a verification procedure using RBF and EBF networks as speaker models and show that the networks are readily applicable to verifying speakers using LP-derived cepstral coefficients as features. Experimental results show that small EBF networks with basis function parameters estimated by the EM algorithm outperform the large RBF networks trained in the conventional approach. The results also show that the equal error rate achieved by the EBF networks is about two-third of that achieved by the vetor quantization (VQ)-based speaker models.**

Index Terms—**EM algorithm, hyperellipsoidal, radial basis functions, speaker verification.**

I. Introduction

RADIAL basis function (RBF) networks have successfully been applied to a wide range of pattern recognition problems [1]–[3]. When used as pattern classifiers, RBF networks represent the posterior probabilities of the training data by a weighted sum of Gaussian basis functions with diagonal covariance matrices. In their most basic form, each diagonal covariance matrix has identical elements controlling the spread of the corresponding RBF unit. As a result, the RBF units are hyperspherical. High recognition accuracy can be achieved when the components of the training vectors (and the unknown test vectors) are independent. If this is not the case, more basis functions are required so that data in the regions covered by each basis function can still be considered to have independent components. It would be beneficial if full covariance matrices could be incorporated into the RBF structure so that complex distributions could be represented without the need for using a large number of basis functions. This paper, therefore, will introduce elliptical basis function (EBF) networks with full covariance matrices in an attempt to enhance the classification capability of conventional RBF networks.

RBF networks can be classified into two categories: normalized and nonnormalized. For the former, the network output is defined as [4], [5]

$$y(\vec{x}) = \frac{\sum_j w_j \phi_j(\vec{x})}{\sum_j \phi_j(\vec{x})}$$

whereas for the latter the output is

$$y(\vec{x}) = \sum_j w_j \phi_j(\vec{x}).$$

In both cases, w_j is the output weight connecting to the jth basis function $\phi_j(\cdot)$. In addition to the architectural difference, these networks have different features and interpretations. For example, the normalized networks can be derived from the least-squares noisy interpolation theory [6] where the input data are assumed to be corrupted by Gaussian noise with variance σ^2. In this case, the network output becomes

$$y(\vec{x}) = \frac{\sum_{n=1}^{N} y_n \exp\left\{-\frac{\|\vec{x}-\vec{x}_n\|^2}{2\sigma^2}\right\}}{\sum_{n=1}^{N} \exp\left\{-\frac{\|\vec{x}-\vec{x}_n\|^2}{2\sigma^2}\right\}}$$

where $\{\vec{x}_n, y_n; n = 1, \cdots, N\}$ are the set of noise-free training data. The idea is to construct a network based on noise-free data so that the network can adapt to a noisy environment by adjusting σ. The normalized networks can also be derived from the theory of kernel regression [7], [8] where the objective is to find a smooth mapping from input space to output space based on a Parzen kernel estimator constructed from the training data. In both cases, the target values y_n become the output weights and there are as many basis function units as the number of input–output pairs. On the other hand, a nonnormalized RBF network can be interpreted as realizing a smooth interpolation function [9], [10]. This is achieved by using the regularization theory through which interpolation functions with large curvature are penalized. In this respect, the weight w_j represents the contribution of the jth basis function to the network output $y(\vec{x})$ when the input pattern $\vec{x}$ is applied.

There have been several studies on using covariance matrices of elliptic shapes for both nonnormalized and normalized RBF networks. For example, Girosi [10] looked at nonnormalized EBF networks from the perspective of function interpolation, and used the information (eigen values) provided by the

Manuscript received February 25, 1998, revised April 2, 1999 and February 25, 2000. This work was supported by the Hong Kong Polytechnic University Grant G-S725 and 1 42.37.A042.

M.-W. Mak is with the Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong.

S.-Y. Kung is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08540 USA.

Publisher Item Identifier S 1045-9227(00)04784-6.

covariance matrices to extract the properties of the function to be approximated. The network's properties were illustrated through fitting the logistic map and a simple two-dimensional interpolation problem. While the networks work very well in these simple problems, it is unclear whether they can be extended to complex real-world problems with high input dimensions. Another approach to learning the basis function parameters is based on competition learning. This includes the clustering algorithm of Musavi *et al.* [11], where the covariance matrices are found by minimizing the overlapping of the nearest neighbors of different classes, and the hyperellipsoidal clustering algorithm of Mao and Jain [12], where the mean vectors and covariance matrices are determined by minimizing the regularized Mahalanobis distance. For normalized EBF networks, Xu *et al.* [13] proposed the alternative model of mixture experts where the basis function parameters and the output weights are simultaneously estimated by the single-loop expectation-maximization (EM) algorithm. Recently, Xu [5] established a connection between the normalized EBF networks and the alternative model of mixture experts by noting that the output weights of a normalized EBF network can be considered as the weighted average of the target values. This connection leads to an EM-based training algorithm and subsequently a fast on-line implementation [5], enabling the EBF parameters and the output weights to be optimized simultaneously in a maximum likelihood framework.

In recent years, the application of the EM algorithm [14] in the estimation of probability density functions has received a great deal of attention (see [15] for a review). The EM algorithm is able to compute the maximum likelihood estimates of the mean vectors and covariance matrices of a Gaussian mixture distribution. Theoretically, the EM algorithm is superior to the combination of the K-means and K-nearest neighbors algorithms as the latter is only capable of estimating the diagonal elements of the covariance matrices. Moreover, the EM algorithm estimates the covariance matrices based on the statistical properties of the data rather than using heuristic methods such as the K-nearest neighbors algorithm. This theoretical choice leads to a more accurate representation of the data being modeled. The EM algorithm has been applied to estimate the parameters of Gaussian mixture models for speaker recognition [16] and phoneme classification [17]. It has also been combined with gradient based learning algorithms to train RBF-like networks. A typical example is the face recognition system proposed by Lin *et al.* [18], where the EM algorithm was applied to estimate the parameters of probabilistic decision-based neural networks, followed by a learning vector quantization (LVQ)-type reinforced and antireinforced learning to fine-tune the decision boundaries and decision thresholds. Other examples include chaotic series prediction [19] and EEG signal classification [20].

While the above studies have used covariance matrices of elliptic shapes to improve performance, they either (1) restrict the matrices to be diagonal [16], [18]–[20], (2) demonstrate the network capability via simple problems [10], or (3) do not provide in-depth comparison among RBF networks, EBF networks with full covariance matrices, and EBF networks with diagonal covariance matrices. This has motivated us to fill this gap and compare these networks via a speaker verification problem in this paper.

In this work, the parameters of the nonnormalized EBF are estimated by the EM algorithm. This is followed by a least squares minimization to determine the output weights. It is interesting to note that the resulting EBF networks can be viewed as the nonnormalized version of the alternative model of mixture experts in [13] as well as the "EM-alternative normalized RBF" networks proposed by Xu [5]. Our approach, however, differs from these models in that it optimizes the basis function parameters before estimating the output weights, whereas these parameters were estimated simultaneously in a maximum likelihood framework in [13], [5].

The organization of this paper is as follows. In Section II, the EBF networks are introduced and various learning algorithms for estimating their parameters are described. Next, the performance of these networks are demonstrated through a speaker verification task. The verification procedure and evaluation method are explained in Section III. The advantages and disadvantages of various training algorithms and network types are then discussed. Finally, we highlight the differences between our work and that of others in Section IV and conclude our discussion in Section V.

II. EBF versus RBF Networks

A. Architecture of EBF Networks

EBF networks can be considered as an extension of the RBF networks [4], [21]. The kth output of an EBF network with I inputs and M function centers has the form

$$y_k(\vec{x}_p) = w_{k0} + \sum_{j=1}^{M} w_{kj}\phi_j(\vec{x}_p) \qquad p = 1, \cdots, N \quad \text{and} \quad k = 1, \cdots, K \tag{1}$$

where

$$\phi_j(\vec{x}_p) = \exp\left\{-\frac{1}{2\gamma_j}\left(\vec{x}_p - \vec{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\vec{x}_p - \vec{\mu}_j\right)\right\} \qquad j = 1, \cdots, M. \tag{2}$$

In (1) and (2), $\vec{x}_p$ is the pth input vector, $\vec{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the mean vector and covariance matrix of the jth basis function respectively, w_{k0} is a bias term, and γ_j is a smoothing parameter controlling the spread of the jth basis function. In this work, γ_j was determined heuristically by

$$\gamma_j = \frac{3}{5}\sum_{k=1}^{5} \|\vec{\mu}_k - \vec{\mu}_j\| \tag{3}$$

where $\vec{\mu}_k$ denotes the kth nearest neighbor of $\vec{\mu}_j$ in the Euclidean sense. Note that this method is similar to the K-nearest neighbor heuristic commonly used in determining the function widths of RBF networks. We have empirically found that using five nearest centers and multiplying the resulting average distance by 3.0 give reasonably good result. However, no attempts have been made to optimize these values. Note also that if the number of centers is less than five, the number of nearest centers used in evaluating γ_j is reduced accordingly.

In matrix form, (1) can be written as $\mathbf{Y} = \mathbf{\Phi W}$ where $\mathbf{Y}$ is an $N \times K$ matrix, $\mathbf{\Phi}$ is an $N \times (M+1)$ matrix, and $\mathbf{W}$ is an $(M+1) \times K$ matrix. The weight matrix $\mathbf{W}$ is the least squares solution of the matrix equation $\mathbf{\Phi W} = \mathbf{D}$, where $\mathbf{D}$ is an $N \times K$ target matrix containing the desired output vectors in its rows. As $\mathbf{\Phi}$ is not a square matrix, one reliable way to find $\mathbf{W}$ is to use the technique of singular value decomposition.

B. Estimation of EBF Parameters

1) K-Means Algorithm and Sample Covariance: The mean vectors and the covariance matrices of an EBF network can be estimated in two steps. In the first step, the K-means algorithm is applied to determine the cluster means and to partition the kth class of the training set, $\mathcal{X}^{(k)}$, into $J^{(k)}$ disjoint clusters $\{\mathcal{X}_j^{(k)}\}_{j=1}^{J^{(k)}}$.[1] Therefore, we estimate the function center $\vec{\mu}_j$ by the sample average $\hat{\vec{\mu}}_j$, i.e.,

$$\vec{\mu}_j \approx \hat{\vec{\mu}}_j = \frac{1}{N_j} \sum_{\vec{x} \in \mathcal{X}_j} \vec{x} \tag{4}$$

where $\vec{x} \in \mathcal{X}_j$ if $\|\vec{x} - \hat{\vec{\mu}}_j\| < \|\vec{x} - \hat{\vec{\mu}}_k\| \; \forall j \neq k$, N_j is the number of samples in the cluster $\mathcal{X}_j$, and $\|\cdot\|$ is the Euclidean norm. In the second step, the covariance matrices are approximated by the sample covariance

$$\mathbf{\Sigma}_j \approx \hat{\mathbf{\Sigma}}_j = \frac{1}{N_j} \sum_{\vec{x} \in \mathcal{X}_j} \left(\vec{x} - \hat{\vec{\mu}}_j\right)\left(\vec{x} - \hat{\vec{\mu}}_j\right)^T. \tag{5}$$

2) The EM Algorithm: Although it has been shown that EBF networks trained in the above two-step approach may give performance superior to RBF networks [22], they may also cause undesirable results when the estimate $\hat{\vec{\mu}}_j$ differs significantly from the true mean $\vec{\mu}_j$. Consequently, the covariance matrix $\hat{\mathbf{\Sigma}}_j$ will no longer be an accurate estimate of the true covariance matrix as an inaccurate mean vector has been used in (5).

To solve this problem, we need an iterative procedure so that the estimated means and the estimated covariance matrices move closer to the maximum likelihood estimates after each iteration. This idea points to the EM algorithm [14] in which the EBF parameters are determined in an iterative fashion. More precisely, the update equations for the mean vectors, full covariance matrices, and mixture coefficients are

$$\vec{\mu}_j^{\text{new}} = \frac{\sum_{\vec{x} \in \mathcal{X}} P^{\text{old}}(j \mid \vec{x}) \vec{x}}{\sum_{\vec{x} \in \mathcal{X}} P^{\text{old}}(j \mid \vec{x})} \tag{6}$$

$$\mathbf{\Sigma}_j^{\text{new}} = \frac{\sum_{\vec{x} \in \mathcal{X}} P^{\text{old}}(j \mid \vec{x}) \left(\vec{x} - \vec{\mu}_j^{\text{new}}\right)\left(\vec{x} - \vec{\mu}_j^{\text{new}}\right)^T}{\sum_{\vec{x} \in \mathcal{X}} P^{\text{old}}(j \mid \vec{x})} \tag{7}$$

$$P^{\text{new}}(j) = \frac{\sum_{\vec{x} \in \mathcal{X}} P^{\text{old}}(j \mid \vec{x})}{\sum_{r=1}^{J} N_r} \tag{8}$$

[1]To simplify the notation, we have dropped the superscript (k) for the remainder of this paper. Therefore, $\mathcal{X}$ denotes the training subset associated with one of the classes and J denotes the number of clusters in that class.

respectively, for all $j = 1, \cdots, J$. In (6), (7), and (8), $P^{\text{old}}(j \mid \vec{x})$ is the posterior probability of the jth cluster, which can be obtained by using Bayes' theorem, yielding

$$P^{\text{old}}(j \mid \vec{x}) = \frac{p(\vec{x} \mid j) P^{\text{old}}(j)}{\sum_k p(\vec{x} \mid k) P^{\text{old}}(k)} \tag{9}$$

where

$$p(\vec{x} \mid j) = \frac{1}{(2\pi)^{\frac{I}{2}} \left|\mathbf{\Sigma}_j^{\text{old}}\right|^{\frac{1}{2}}} \times \exp\left\{-\frac{1}{2}\left(\vec{x} - \vec{\mu}_j^{\text{old}}\right)^T \left(\mathbf{\Sigma}_j^{\text{old}}\right)^{-1} \left(\vec{x} - \vec{\mu}_j^{\text{old}}\right)\right\} \tag{10}$$

is the probability density function of the jth cluster. When the covariance matrices are diagonal, (7) and (10) become

$$\left(\sigma_{ji}^{\text{new}}\right)^2 = \frac{\sum_{\vec{x} \in \mathcal{X}} P^{\text{old}}(j \mid \vec{x}) \left(x_i - \mu_{ji}^{\text{new}}\right)^2}{\sum_{\vec{x} \in \mathcal{X}} P^{\text{old}}(j \mid \vec{x})} \quad i = 1, \cdots, I \tag{11}$$

and

$$p(\vec{x} \mid j) = \frac{1}{(2\pi)^{\frac{I}{2}} \prod_{i=1}^{I} \sigma_{ji}^{\text{old}}} \exp\left\{-\frac{1}{2} \sum_{i=1}^{I} \frac{\left(x_i - \mu_{ji}^{\text{old}}\right)^2}{\left(\sigma_{ji}^{\text{old}}\right)^2}\right\} \tag{12}$$

respectively. Note that if $P^{\text{old}}(j \mid \vec{x})$ is equal to 1.0 for all $\vec{x} \in \mathcal{X}_j$ and is equal to 0.0 otherwise, (6) and (7) will be reduced to (4) and (5), respectively. Therefore, the K-means algorithm and the sample covariance are special cases of the EM algorithm.

The EM algorithm has several advantages over the gradient-based approach in estimating model parameters even though there is a mathematical connection between them [23]. First, the EM algorithm has low computational overheads. Second, probability constraints on the estimated parameters can be satisfied automatically in EM, while the gradient-based algorithms require additional checks to ensure that the constraints are satisfied, e.g., addition of penalty terms in the error function. Third, the EM algorithm guarantees monotonic convergence without the need to specify a learning rate.

III. Application to Speaker Verification

We have applied the EBF and RBF networks trained with the algorithms mentioned in the previous section to a speaker verification task. Speaker verification is to verify whether the voice of a claimant matches the voice of the claimed identity. This technology makes access control by the human voice possible. This section describes the speaker verification experiments and provides an in-depth comparison between different types of networks and learning algorithms.

A. Speech Corpus and Feature Extraction

The TIMIT corpus was used in the experiments. TIMIT is a phonetically balanced, continuous speech corpus containing 630 speakers separated into eight dialect regions. In the experiments, we used 258 speakers (186 male, 72 female) from the first four dialect regions of the corpus. These speakers were

divided into four sets: speaker set (76 speakers from dialect region 2), antispeaker set (38 speakers from region 1), pseudoimpostor set (68 speakers from region 4), and impostor set (76 speakers from region 3). The purpose of these sets will be explained in the next few sections.

LP-derived cepstral coefficients were used as feature vectors.[2] For each sentence, the silent regions were removed by using the information provided by the phonetic transcription files (.phn) of the corpus. The remaining signals were pre-emphasized by a filter with transfer function $H(z) = 1 - 0.95z^{-1}$. For every 14 ms, 12th-order LP-derived cepstral coefficients were computed using a 28 ms Hamming window.

B. Enrollment

Each speaker in the speaker set was assigned a personalized network (RBF or EBF) modeling the characteristics of his/her own voice. For each network, the feature vectors derived from the SA and SX sentence sets were used for training. Each network was trained to recognize the data derived from two classes—speaker class and antispeaker class. The former was derived from the speaker set while the latter from the antispeaker set. Therefore, each network was composed of 12 inputs (12th-order cepstral coefficients were used as features), varied numbers of hidden nodes, and two outputs, with each output representing one class.

The enrollment procedure consists of five steps. These are described as follows.

Step 1) Apply the K-means algorithm to the cepstral vectors of the speaker being enrolled. The resulting centers are referred to as the speaker centers.

Step 2) Apply the K-means algorithm to the cepstral vectors of all antispeakers in the antispeaker set to obtain a pool of function centers. These centers are referred to as the anticenters.

Step 3)

a) If the network is an EBF one and its basis function parameters are to be estimated by sample covariance, apply (5) to obtain the function widths corresponding to the speaker centers using the cepstral vectors of the speaker as $\vec{x}$ and the speaker centers obtained in Step 1) as $\vec{\mu}_j$. Similarly, (5) is applied to the cepstral vectors of the antispeakers to obtain the widths corresponding to the anticenters. Then, go to Step 4).

b) If the network is an RBF one, apply the K-nearest neighbors algorithm (with $K = 2$) to the anticenters to obtain the function widths corresponding to the anticenters. The function widths corresponding to the speaker centers are obtained similarly. Then, go to Step 4).

c) If the network is an EBF one whose basis function parameters are to be estimated by the EM algorithm, apply the K-nearest neighbors algorithm to the speaker centers and anti-centers separately as in Step 3b) above to initialize the function widths. Then, apply (6) to (12) repeatedly using the centers obtained in Steps 1) and 2) as the initial values of $\vec{\mu}_j^{\text{old}}$ and using the function widths obtained by the K-nearest neighbors algorithm as the initial values of Σ_j^{old}. Then, go to Step 4).

Step 4) Compute γ_j in (2) according to (3) and compute the matrix Φ. Apply singular value decomposition to find the output weights $\mathbf{W}$.

Step 5) Determine the decision threshold according to Section III-D.

Note that the above clustering procedure in Steps 1) to 3) was applied to the speaker class and the antispeaker class independently, which is different from the conventional way of training RBF networks where the K-means algorithm is applied to the data from all classes. Our approach has computational and storage advantages over the conventional one because all speakers share the same set of "anticenters" which only need to be determined once. In the conventional approach, however, the anticenters and their associated covariance matrices have to be evaluated for each speaker, resulting in a much longer enrollment time. The substantial saving in computation time also enables us to use a large number of antispeakers (38 in this study) to improve the capability of the networks in modeling impostors' speech.

C. Verification

As a 1-of-K coding scheme was used and the output units are linear, the network outputs are estimates of the *a posteriori* probabilities, i.e., $P(C_k \mid \vec{x})$ where C_k and $\vec{x}$ represent the kth class and an unknown input vector, respectively. The average of each output over the whole training set is an estimate of the prior probability $P(C_k)$. Therefore, if the number of patterns in the training set is not evenly distributed among the classes, the network outputs will demonstrate a bias toward those classes with a larger proportion of patterns. For instance, in a two-class problem (i.e., $K = 2$) where the prior probability of one class is significantly less than that of the other, say $P(C_1) \ll P(C_2)$, it is likely that the output $y_1(\vec{x})$ is less than the output $y_2(\vec{x})$ irrespective of the class that $\vec{x}$ belongs to.

In the experiments, each speaker contributes the same number of sentences for training. As a result, the ratio of training vectors between the speaker class and the antispeaker class is about $1 : 38$. This is because each network uses one speaker from the speaker set and 38 speakers from the antispeaker set for training. The network will favor the antispeaker class during verification by always giving outputs which are close to one for the antispeaker class and close to zero for the speaker class. Weighting the error function according to the *a priori* probabilities is one way to circumvent this problem [25]. Alternatively, we can scale the outputs during verification so that the new average outputs are approximately equal to 0.5 for both classes. This can be achieved by multiplying the output $y_k(\vec{x})$ by $1/(2P(C_k))$. Specifically, we computed the scaled output

$$\tilde{y}_k(\vec{x}) = \frac{1}{2} \cdot \frac{y_k(\vec{x})}{P(C_k)} \quad k = 1, 2 \tag{13}$$

[2]Atal [24] found that cepstral coefficients are one of the best features for speaker recognition.

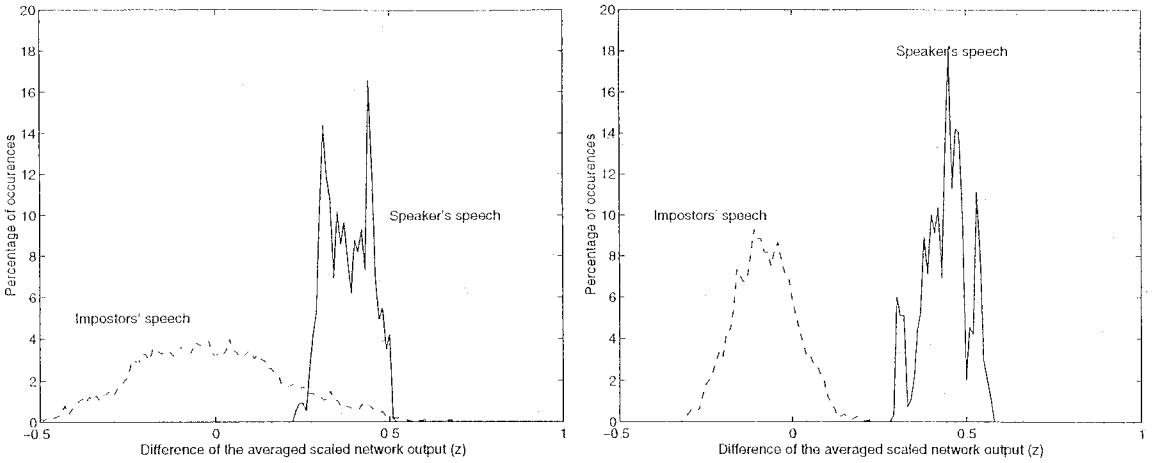


Fig. 1. The distributions of z corresponding to (a) an RBF network and (b) an EBF network. Both networks contain 12 centers, four from the speaker and eight from the antispeaker.

so that $(1/N')\sum_{x\in\mathcal{X}'}\tilde{y}_k(\vec{x}) \approx 0.5$, where N' denotes the number of patterns in the training set $\mathcal{X}'$. A simple way to estimate the prior probability $P(C_k)$ is to divide the number of patterns in class C_k by the total number of patterns in the training set.

During verification, a vector sequence $\mathcal{T} = [\vec{x}_1, \cdots, \vec{x}_T]$ corresponding to an utterance spoken by an unknown speaker was fed into the network. Then we computed the scaled average outputs

$$z_k = \frac{1}{T}\sum_{\vec{x}\in\mathcal{T}} \frac{\exp\{\tilde{y}_k(\vec{x})\}}{\exp\{\tilde{y}_1(\vec{x})\} + \exp\{\tilde{y}_2(\vec{x})\}} \quad k = 1, 2 \qquad (14)$$

corresponding to the speaker and antispeaker classes. Note that we have made use of the softmax function inside the summation of (14). The purpose is to ensure that z_k is in the range $[0, 1]$ and that $\sum_k z_k = 1$, thereby preventing any extreme value of $\tilde{y}_k$ from dominating the average outputs. Verification decisions were based on the criterion

$$\text{If } z = z_1 - z_2 \begin{cases} >\zeta : \text{accept the unknown speaker} \\ \leq\zeta : \text{reject the unknown speaker} \end{cases} \qquad (15)$$

where $\zeta \in [-1, 1]$ is a threshold controlling the false rejection rate (FRR) and the false acceptance rate (FAR). For example, if ζ is set to 1.0, the unknown speaker will likely be rejected, resulting in a high FRR but a low FAR.

The method mentioned above can be used to verify unknown speakers based on a single utterance or multiple utterances from the test set. However, this will only give a single decision for each utterance—accept or reject. As a result, a large number of test utterances will be required if we want to increase the resolution of the error rates. To address this problem, we concatenated the feature vectors derived from the utterances of an unknown speaker to form a test sequence $\mathcal{T}' = [\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_{T'}]$. The sequence was then divided into a number of overlapping segments containing 200 consecutive vectors (2.8 s of speech), i.e., T in (14) is equal to 200. A verification decision was made for every segment. After each verification decision, a window covering 200 consecutive vectors was moved forward by one vector in the sequence and the verification procedure was repeated. The error rate is the proportion of incorrect verification decisions to the total number of verification decisions. By adopting this approach, about 500 and 40 000 decisions would be made to determine the FRR and FAR, respectively, for each speaker in the speaker set.

We can investigate the effectiveness of this approach by examining the network output. Fig. 1(a) depicts the distributions of the difference between the two outputs, z [see (14)], of the RBF network associated with the speaker "faem0." Fig. 1(b) shows the corresponding distributions of an EBF network whose basis function parameters were determined by the EM algorithm. The distributions were obtained by feeding the cepstral vectors derived from "faem0" (speaker's speech) and from the impostor set (impostors' speech) to the networks. The results show that both networks are able to distinguish the voices of the speaker from that of the impostors as their voices produce two distinguishable distributions. However, it is evident that for the EBF network, the distribution corresponding to impostors' speech exhibits a smaller spread, making the two distributions more distinguishable (less overlapped). As a result, the EBF network has a lower FAR as compared to the RBF network for the same threshold, as shown in Fig. 2. Fig. 2 also shows that the equal error rate (the crossing point of FAR and FRR) is smaller for the EBF network.

D. Decision Thresholds

The decision threshold ζ for each network was determined during the enrollment phase. After a network has been trained, the verification procedure as described in Section III-C was applied. However, instead of using the speech of an unknown speaker, the feature vectors of pseudoimpostors in the pseudoimpostor set were used. The threshold was adjusted between the range $[-1, +1]$ until the FAR fell below a predefined value. In this work, the predefined FAR was set to 2%.

Once the threshold value has been found, the false rejection rate corresponding to each speaker was obtained by presenting

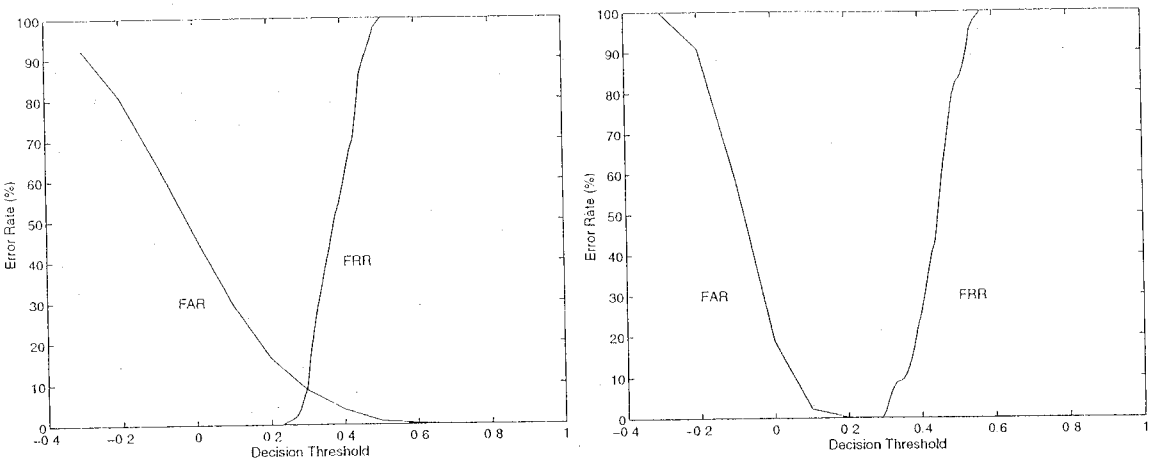


Fig. 2. FAR and FRR versus the decision threshold of (a) an RBF network and (b) an EBF network. Both networks contain 12 centers, four from the speaker and eight from the antispeakers.

TABLE I
ABBREVIATIONS OF EXPERIMENT TITLES, NETWORK TYPES, AND ALGORITHMS USED IN ESTIMATING THE BASIS FUNCTION PARAMETERS

Exp. Abbr.	Network Type	Clustering Algorithms
R	RBF	K-means and K-nearest neighbors
EC	EBF	K-means and sample covariance
EED	EBF	EM with diag. covariance matrices
EEF	EBF	EM with full covariance matrices

the SI sentence set of the speaker to his/her own network. The false acceptance rate was obtained by feeding the SI sentence set of all impostors (from the impostor set) into the network.

E. Verification Experiments and Results

We have tried various combinations of network types (RBF and EBF) and learning algorithms (K-means, K-nearest neighbors, sample covariance, and EM). Table I summarizes the verification experiments we have conducted.

Table II summarizes the false acceptance rates (FARs), false rejection rates (FRRs), and equal error rates (EERs) for different network types, network sizes, and learning algorithms. The equal error rates were obtained by adjusting the thresholds during verification until FAR is equal to FRR. All error rates in Table II were based on the average of 76 speakers in the speaker set.

The results of Table II demonstrate the superiority of the EBF networks over the RBF networks. In particular, Table II shows that the equal error rate of the smallest EBF network (*EEF* with ten centers) is 0.04%, while that of the largest RBF network (*R* with 24 centers) is 8.06%. This illustrates that the full covariance matrices of the EBF networks are capable of providing a better representation of the feature vectors, even though their number is smaller.

We can see from Table II that for all size of network, the EBF networks trained with the EM algorithm (*EEF*) attain a lower equal error rate as compared to the EBF networks trained in sample covariance (*EC*). This suggests that the mean vectors and the full covariance matrices found by the EM algorithm are better than those found by the sample covariance. This agrees with our previous conjecture that the sample covariance (5) may give poorer estimates of the true covariance matrices. Table II also allows us to compare the performance of networks with different numbers of speaker centers. The last four rows of Table II show that the equal error rates are generally smaller for networks with a larger number of speaker centers. However, the optimal numbers of speaker centers and anticenters remain unknown.

The results also show that the performance of EBF networks with diagonal covariance matrices (*EED*) is poorer than that of the EBF networks with full covariance matrices (*EEC* and *EEF*). This is because the principal axes of the basis functions with diagonal covariance matrices are parallel to that of the feature space. This restriction reduces the capability of the basis functions in modeling the statistical characteristics of the feature vectors. Despite this limitation, their performance is still better than the RBF networks where the width of each basis function must be the same. This restriction further reduces the flexibility of the RBF networks in modeling the statistical characteristics of the feature vectors, resulting in higher error rates.

Table III shows the error rates obtained by applying vector quantization (VQ) speaker models to the same set of data. The VQ models were obtained by the classical LBG algorithm [26]. There are two main differences between the VQ speaker models and the RBF- or EBF-speaker models proposed in this paper. First, the VQ models do not require antispeakers' data during enrollment (parameter estimation). Second, VQ adopts the so-called hard partitioning approach where each data is assigned to one of the known clusters; whereas the neural models adopt the soft partitioning approach via the EM training procedure. Therefore, in the neural models, each data is assigned to all clusters but with different degree of membership. Tables II and III reveals that the error rate obtained by VQ is comparable to that obtained by EBF network with diagonal covariance matrices (*EED*). However, the EBF networks with full covariance matrices (*EEF* and *EC*) achieve a considerably lower equal error rate as compared to VQ, even for VQ models with a large codebook size.

TABLE II
FAR's, FRR's, AND EER's (IN %) FOR NETWORKS WITH VARIOUS NUMBERS OF CENTERS. EACH NETWORK CONTAINS TWO TO 16 CENTERS CONTRIBUTED FROM THE CORRESPONDING SPEAKER AND THE REST ARE FROM THE ANTISPEAKERS. FOR EXAMPLE, THE NETWORK WITH TEN CENTERS HAS EIGHT CENTERS FROM THE CORRESPONDING SPEAKER AND TWO FROM THE ANTISPEAKERS, I.E., (8 + 2) CENTERS

Abbr.	Number of Centers per Network											
	10 (8+2)			12 (8+4)			16 (8+8)			24 (8+16)		
	FAR	FRR	**EER**	FAR	FRR	**EER**	FAR	FRR	**EER**	FAR	FRR	**EER**
R	29.56	58.70	**19.15**	29.67	57.89	**19.30**	45.38	31.16	**8.27**	54.20	23.88	**8.06**
EC	52.77	0.00	**0.44**	13.75	0.00	**0.06**	5.66	0.00	**0.04**	0.01	76.85	**0.24**
EED	42.80	0.00	**0.27**	26.48	0.00	**0.17**	11.10	0.43	**0.22**	2.79	4.48	**0.14**
EEF	21.25	0.00	**0.04**	20.00	0.00	**0.03**	8.32	0.00	**0.03**	3.08	1.29	**0.04**
	Number of Centers per Network											
	10 (2+8)			12 (4+8)			16 (8+8)			24 (16+8)		
	FAR	FRR	**EER**	FAR	FRR	**EER**	FAR	FRR	**EER**	FAR	FRR	**EER**
R	81.29	13.16	**13.02**	46.47	34.74	**11.87**	45.38	31.16	**8.27**	74.42	10.75	**9.64**
EC	0.46	14.99	**0.44**	3.75	0.49	**0.08**	5.66	0.00	**0.04**	6.77	0.00	**0.02**
EED	1.25	50.54	**1.02**	6.48	8.16	**0.56**	11.00	0.43	**0.22**	7.44	0.00	**0.13**
EEF	3.52	6.74	**0.37**	4.95	0.75	**0.05**	8.32	0.00	**0.03**	7.39	0.00	**0.02**

TABLE III
FAR's, FRR's, AND EER's (IN %) USING VQ SPEAKER MODELS

Codebook size	FAR	FRR	EER
2	0.85	66.88	**11.70**
4	1.21	39.29	**5.21**
8	1.30	20.17	**2.65**
16	1.41	6.51	**1.14**
32	1.44	5.29	**0.78**
64	2.30	1.79	**0.55**
128	2.30	0.97	**0.45**
256	2.26	0.70	**0.36**
512	2.21	1.12	**0.35**

As the numbers of free parameters of EBF and RBF networks with equal number of basis functions could be rather different, one may argue that the superiority of EBF networks is due to the large number of free parameters that they possess. Therefore, it makes sense to compare their recognition performance with respect to the number of free parameters instead of the number of function centers. Table IV lists the error rates for the RBF and EBF networks with different network sizes but with similar numbers of free parameters. It shows that EBF networks with full covariance matrices trained with the EM algorithm achieve the lowest equal error rate. This result demonstrates the capability of the EM algorithm and the advantage of using full covariance matrices in the basis functions.

In terms of FAR and FRR, Table IV shows that the VQ approach is the best. However, one should bear in mind that these values are highly dependent on the decision thresholds. A closer look at the error rates of individual speakers reveals that for the EBF networks, some speakers have a very high FRR but a very low FAR, or vice versa. This suggests that the thresholds have not been optimized, and therefore there is plenty of room for reducing the FAR's and FRR's corresponding to these networks. On the other hand, the EER's were obtained by adjusting the thresholds during verification to equalize the FAR and FRR of each speaker. Although this adjustment is impractical in real systems, the EER's indicate the potential capability of the networks. This capability will become achievable once better thresholds have been found. We have recently proposed a threshold determination method [27] and are currently extending it to a two-stage scoring method for speaker verification [28]. These methods should, in principle, enable us to find better thresholds for the networks.

In addition to the low-level features such as spectral characteristics of speech, human can also recognize the voice of individuals based on some high-level features such as dialects and speaking style. It is interesting to know whether the networks have similar capability. In other words, we would like to answer the question: Does the difference in dialect play a role in helping the networks to recognize human voices? To this end, we select speakers from the speaker set as impostors so that the speakers and impostors have the same dialect. More specifically, for each speaker in the speaker set (dialect region 2), we selected 75 impostors from the same dialect region. The experiments that leads to the columns under "Different Dialect" in Table IV were repeated, and the results are shown in the columns under "Same Dialect" in Table IV. The results show that the dialect of impostors has little effect on the error rate, and more importantly the effect is not consistent across different network types. For example, using impostors with dialect identical to that of the true speakers produces lower error rates in experiments *R* and *EEF*, whereas the other experiments show the opposite. This suggests that the networks do not recognize speakers based on their dialects. This result is actually not surprising because the networks are designed to recognize the spectral features (i.e., shape of the vocal tract) in a frame-by-frame basis; they have never been trained to recognize high level features in which contextual representation is likely to be more important than the local spectral representation.

IV. RELATED WORK

By means of a hand-written character recognition task and a vowel classification task, Nowlan [29] showed that EBF networks with diagonal covariance matrices give a poorer performance as compared to the RBF networks, which does not agree with our results. Nowlan explained that the EBF networks are inferior because they cannot represent rapidly changing density

TABLE IV

ERROR RATES (IN %) FOR NETWORKS WITH COMPARABLE NUMBERS OF PARAMETERS. THE SECOND COLUMN LISTS THE TOTAL NUMBER OF CENTERS PER NETWORK, INCLUDING SPEAKER CENTERS AND ANTICENTERS. FOR EXAMPLE, $(12 + 49)$ MEANS THAT THE NETWORKS HAVE 12 SPEAKER CENTERS AND 49 ANTICENTERS. THE RATIO BETWEEN SPEAKER CENTERS AND ANTICENTERS IS APPROXIMATELY EQUAL TO FOUR. THE COLUMNS UNDER "DIFFERENT DIALECT" REPRESENT THE ERROR RATES OBTAINED BY USING IMPOSTORS OF DIFFERENT DIALECT WITH RESPECT TO THE TRUE SPEAKERS, WHEREAS THE TERM "SAME DIALECT" MEANS THAT THE SPEAKERS AND IMPOSTORS ARE OF THE SAME DIALECT

Exp. Title	No. of Centers	No. of Parameters	Different Dialect			Same Dialect		
			FAR	FRR	EER	FAR	FRR	EER
R	61 (12 + 49)	917	20.72	53.93	**7.46**	20.08	53.93	**7.28**
EC	10 (2 + 8)	922	0.46	14.99	**0.44**	0.47	14.99	**0.46**
EED	35 (7 + 28)	912	1.78	15.07	**0.47**	1.86	15.07	**0.49**
EEF	10 (2 + 8)	922	3.52	6.74	**0.37**	3.01	6.74	**0.30**
VQ64	64	768	2.30	1.79	**0.55**	2.52	1.79	**0.57**
VQ128	128	1536	2.30	0.97	**0.45**	2.47	0.97	**0.46**

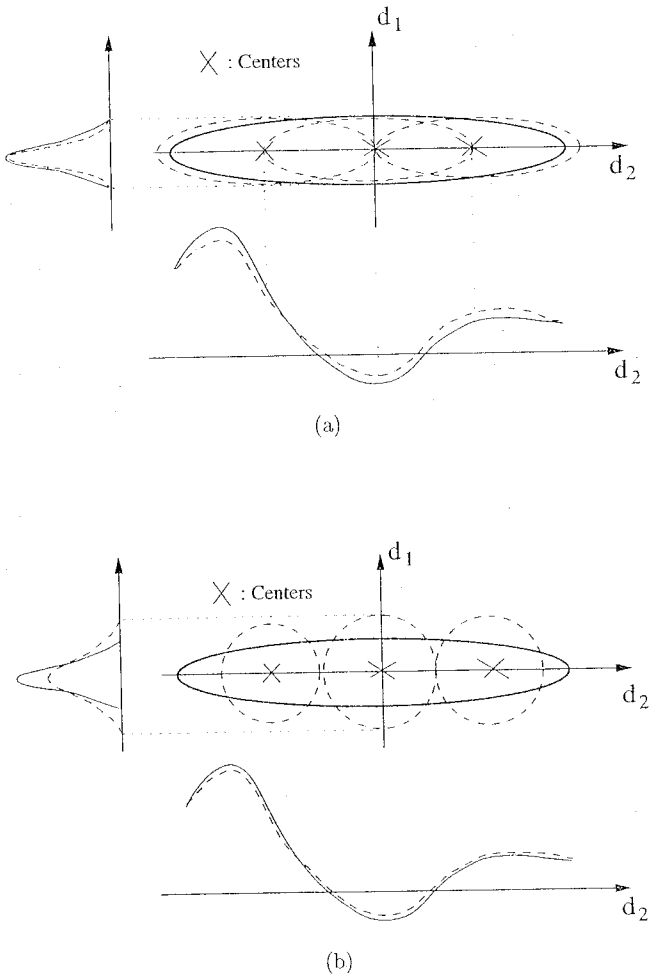


Fig. 3. Equal probability contours (dashed ellipses and circles) of (a) elliptical Gaussians and (b) spherical Gaussians. The solid ellipse represents the data distribution. The solid and dashed graphs to the left of the data distribution are respectively the projections onto the d_1 axis of a function to be interpolated and the approximation to the function. The graphs underneath the distribution represent the corresponding projections onto the d_2 axis (after [29]).

functions accurately. This argument is supported by a hypothetical situation where EBF and RBF networks were used to interpolate a function with two independent variables. The training data exhibit small variance along one input dimension (d_1) and much larger variance in another dimension (d_2). The hypothetical situation is redrawn in Fig. 3 for ease of discussion. Fig. 3 demonstrates that although the interpolation along d_2 is poorer for the EBF network, the interpolation exhibits a larger error along d_1 for the RBF network. This is because should the radii of the spherical units be reduced to better fit the data along d_1, the interpolation error along d_2 will be increased. Therefore, considering the interpolation error along a single dimension may not allow us to see the whole picture.

As the ultimate objective of the present study is classification, not interpolation, it may be more intuitive to compare the classification accuracy than to compare the output function produced by the networks. Our results clearly demonstrate that the EBF networks with diagonal covariance matrices outperform the RBF networks (0.47% versus 7.46%, see Table IV).

No comparison between EBF networks that use full covariance matrices and that use diagonal ones has been made in [29], whereas we have done this empirically in this study and showed that the performance of the former is significantly better. We have also suggested to use a noniterative algorithm (the sample covariance) to estimate the full covariance matrices and showed that this approach degrades the performance (in terms of error rates) slightly. Bear in mind that the sample covariance can determine the matrices in one pass, whereas the EM algorithm requires a number of iterations to obtain the maximum likelihood estimates. Therefore, sample covariance is a viable alternative to the EM algorithm if training time is an important issue.

V. CONCLUSION

In this paper, we proposed to apply the EM algorithm to estimate the basis function parameters of elliptical basis function networks. The proposed learning scheme enables the maximum likelihood estimates of the EBF parameters to be found, resulting in a higher recognition accuracy. We have evaluated and compared the performance of the EBF and RBF networks through a series of text-independent speaker verification experiments. Several conclusions can be drawn from the results of these experiments. First, we have found that for the same number of function centers, EBF networks with full covariance matrices trained with the EM algorithm outperform the ones whose basis function parameters are estimated by sample covariance. Second, RBF networks are found to be the poorest performers in terms of verification accuracy. Finally, this study

has shown that when the number of free parameters are comparable, EBF networks with full covariance matrices achieve the lowest equal error rate.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments, which were very useful in improving the manuscript.

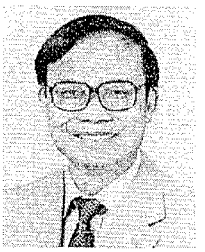
REFERENCES

[1] S. Renals, "Radial basis function for speech pattern classification," *Electron. Lett.*, vol. 25, no. 7, pp. 437–439, 1989.
[2] Y. Lee, "Handwritten digit recognition using K-nearest-neighbor, radial basis function, and backpropagation networks," *Neural Computing*, vol. 3, no. 3, pp. 440–449, 1991.
[3] M. W. Mak, W. G. Allen, and G. G. Sexton, "Speaker identification using multilayer perceptrons and radial basis function networks," *Neurocomput.*, vol. 6, pp. 99–117, 1994.
[4] J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units," *Neural Comput.*, vol. 1, pp. 281–194, 1989.
[5] L. Xu, "RBF nets, mixture experts, and Bayesian Ying-Yang learning," *Neurocomputing*, vol. 19, pp. 223–257, 1998.
[6] A. R. Webb, "Functional approximation by feedforward networks: A least-squares approach to generalization," *IEEE Trans. Neural Networks*, vol. 5, pp. 363–371, 1994.
[7] H. Schiøler and U. Hartmann, "Mapping neural network derived from the Parzen window estimator," *Neural Networks*, vol. 5, no. 6, pp. 903–909, 1992.
[8] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, pp. 109–118, 1990.
[9] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.
[10] F. Girosi, "Some extensions of radial basis functions and their applications in artificial intelligence," *Comput. Math. Applicat.*, vol. 24, no. 12, pp. 61–80, 1992.
[11] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels, "On the training of radial basis function classifiers," *Neural Networks*, vol. 5, pp. 595–603, 1992.
[12] J. Mao and A. K. Jain, "A self-organizing network for hyperellipsoidal clustering (HEC)," *IEEE Trans. Neural Networks*, vol. 7, pp. 16–29, Jan. 1996.
[13] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," in *Advances in Neural Information Processing Systems 7*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. Cambridge, MA: MIT Press, 1995, pp. 633–640.
[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, ser. B, vol. 39, no. 1, pp. 1–38, 1977.
[15] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Mag.*, vol. 13, pp. 47–60, Nov. 1996.
[16] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 72–83, Jan. 1995.
[17] H. G. C. Tråvén, "A neural network approach to statistical pattern classification by semiparametric estimation of probability density functions," *IEEE Tran. Neural Networks*, vol. 2, pp. 366–377, May 1991.
[18] S. H. Lin, S. Y. Kung, and L. J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Trans. Neural Networks*, vol. 8, pp. 114–132, Jan. 1997.
[19] A. Ukrainec and S. Haykin, "Signal processing with radial basis function networks using expectation maximization algorithm clustering," *Proc. SPIE*, vol. 1565, pp. 529–539, 1991.
[20] L. Tarassenko and S. Roberts, "Supervised and unsupervised learning in radial basis function classifiers," *Proc. Inst. Elect. Eng., Vis. Image Signal Processing*, vol. 141, no. 4, Aug. 1994.
[21] D. S. Broomhead and D. Lowe, "Multivariable function interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.
[22] M. W. Mak, "Text-independent speaker verification over a telephone network by radial basis function networks," in *Int. Symp. Multitechnol. Inform. Processing*, Taiwan, R.O.C., 1996, pp. 145–150.
[23] L. Xu and M. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Comput.*, vol. 8, no. 8, pp. 129–151, 1996.
[24] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, no. 6, pp. 1304–1312, 1974.
[25] D. Lowe and A. R. Webb, "Exploiting prior knowledge in network optimization: An illustration from medical prognosis," *Network: Comput. Neural Syst.*, vol. 1, pp. 299–323, 1990.
[26] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COMM-28, pp. 84–95, Jan. 1980.
[27] W. D. Zhang, K. K. Yiu, M. W. Mak, C. K. Li, and M. X. He, "*A priori* threshold determination for phrase-prompted speaker verification," in *Proc. Eurospeech'99*, vol. 2, Sept. 1999, pp. 775–778.
[28] W. D. Zhang, M. W. Mak, and M. X. He, "A two-stage scoring method combining world and cohort model for speaker verification," in *Proc. ICASSP'2000*, vol. 2, June 2000, pp. 1193–1196.
[29] S. J. Nowlan, "Soft Competitive Adaptation: Neural Network Learning Algorithms Based on Fitting Statistical Mixtures," Ph.D. dissertation, School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, 1991.

Man-Wai Mak (S'91–M'93) received the B.Eng. (Hons.) degree in electronic engineering from Newcastle upon Tyne Polytechnic, U.K., in 1989 and the Ph.D. degree in electronic engineering from the University of Northumbria at Newcastle, U.K., in 1993.

He was a Research Assistant at the University of Northumbria at Newcastle from 1990 to 1993. He joined the Department of Electronic Engineering at the Hong Kong Polytechnic University as a Lecturer in 1993 and as an Assistant Professor in 1995. His research interests include neural networks, speaker verification, and audio processing.

Dr. Mak has been an executive committee member of the IEEE Hong Kong Section Computer Chapter since 1995. He is currently the Secretary of the IEEE Hong Kong Section Computer Chapter.. He is currently the Secretary of the IEEE Hong Kong Section Computer Chapter.



Sun-Yuan Kung (S'74–M'78–SM'84–F'88) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

In 1974, he was an Associate Engineer with Amdahl Corporation, Sunnyvale, CA. From 1977 to 1987, he was a Professor of electrical engineering-systems, the University of Southern California, Los Angeles. Since 1987, he has been a Professor of electrical engineering, Princeton University, Princeton, NJ. He has authored more than 300 technical publications, including three books *VLSI Array Processors*, (Englewood Cliffs, NJ: Prentice-Hall, 1988) (with Russian and Chinese translations), *Digital Neural Networks*, (Englewood Cliffs, NJ: Prentice-Hall, 1993), and *Principal Component Neural Networks*, (New York: Wiley, 1996).

Dr. Kung was the recipient of the 1992 IEEE Signal Processing Society's Technical Achievement Award for his contributions on "parallel processing and neural-network algorithms for signal processing." He was appointed as an IEEE-SP Distinguished Lecturer in 1994. He received 1996 IEEE Signal Processing Society's Best Paper Award. He was a recipient of the IEEE Third Millennium Medal in 2000. Since 1990, he has served as an Editor-In-Chief of *Journal of VLSI Signal Processing Systems*. He served as a founding member and General Chairman of various international conferences, including IEEE Workshops on VLSI Signal Processing in 1982 and 1986 (Los Angeles), International Conference on Application Specific Array Processors in 1990 (Princeton) and 1991 (Barcelona), and IEEE Workshops on Neural Networks and Signal Processing in 1991 (Princeton), 1992 (Copenhagen) and 1998 (Cambridge, U.K.), the First IEEE Workshops on Multimedia Signal Processing in 1997 (Princeton), and International Computer Symposium in 1998 (Taiwan).