Elliptical Basis Function Networks and Radial Basis Function Networks for Speaker Verification: A Comparative Study

M. W. Mak and C. K. Li

Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong.
Email: enmwmak@polyu.edu.hk

ABSTRACT

It is well known that radial basis function (RBF) networks require a large number of function centers if the data to be modeled contain clusters with complicated shape. This paper proposes to overcome this problem by incorporating full covariance matrices into the RBF structure and to use the expectation-maximization (EM) algorithm to estimate the network parameters. The resulting networks, referred to as the elliptical basis function (EBF) networks, are applied to text-independent speaker verification. Experimental evaluations based on 258 speakers of the TIMIT corpus show that smaller size EBF networks with basis function parameters determined by the EM algorithm outperform the large RBF networks trained by the conventional approach.

1. INTRODUCTION

The conventional approach [1] to estimating the parameters of radial basis function (RBF) networks involves an unsupervised stage (K-means and K-nearest neighbors) to find the center positions and to determine the function widths, followed by a supervised stage to compute the output weights. This two-stage approach, however, may lead to sub-optimal performance when the input vectors contain correlated components. The RBF networks solve this problem by using a large number of basis functions so that data in regions covered by each basis function can still be considered to have uncorrelated components. It would be beneficial if full covariance matrices can be incorporated into the RBF structure so that complex distributions could be represented without the need of using a large number of basis functions.

Recently, the application of the expectation maximization (EM) algorithm [2] in the estimation of probability density functions has received a great deal of attention. 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 K-means and K-nearest neighbor algorithms as the latter has an intrinsic limitation that all the function widths are identical for all input dimensions.

The EM algorithm has been applied to estimate the parameters of Gaussian mixture models for speaker recognition [3] and phoneme classification [4]. It has also been combined with gradient based learning algorithms to train RBF-like networks [5]. Most of these studies, however, used diagonal covariance matrices in the networks. In this paper, we propose to use full covariance matrices in the basis functions, resulting in elliptical basis function (EBF) networks. To the best of our knowledge, there have been no studies in which the recognition performance of EBF networks with full covariance matrices is compared with that of RBF networks. This has motivated us to compare these two types of networks through a series of speaker verification experiments in this paper.

The paper is organized as follows. In Section 2, the EBF networks and their training algorithms are introduced. Section 3 explains the speaker verification experiments through which the performance of EBF and RBF networks is compared in Section 4. Finally, a conclusion is drawn in Section 5.

2. EBF NETWORKS VS. RBF NETWORKS

2.1. Architecture of EBF networks

EBF networks can be considered as an extension of RBF networks. The kth output ($k = 1, \ldots, K$) of an EBF network with I inputs and M function centers has the form

$$y_k(\vec{x}_p) = w_{k0} + \sum_{m=1}^{M} w_{km}\phi_m(\vec{x}_p) \quad p = 1, \ldots, N \quad (1)$$

This project was supported by The H.K. Polytechnic University Grant No. G-S472.

where $\phi_m(\vec{x}_p) = \exp\left\{\frac{-1}{2\gamma_m}(\vec{x}_p - \vec{\mu}_m)^T \mathbf{\Sigma}_m^{-1}(\vec{x}_p - \vec{\mu}_m)\right\}$. In (1), $\vec{x}_p$ is the pth input vector, $\vec{\mu}_m$ and $\mathbf{\Sigma}_m$ are the mean vector and covariance matrix of the mth basis function respectively, w_{k0} is a bias term, and γ_m is a smoothing parameter controlling the spread of the mth basis function. In this work, γ_m was determined heuristically by

$$\gamma_m = \frac{3}{5}\sum_{k=1}^{5}\|\vec{\mu}_k - \vec{\mu}_m\| \tag{2}$$

where $\vec{\mu}_k$ denotes the k-th nearest neighbor of $\vec{\mu}_m$ in the Euclidean sense. 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 5, the number of nearest centers used in evaluating γ_m is reduced accordingly.

In matrix form, (1) can be written as $\mathbf{Y} = \mathbf{\Phi}\mathbf{W}$, 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, where N is the number of training patterns. Once the elements of $\mathbf{\Phi}$ are known, the weight matrix $\mathbf{W}$ can be determined by a least squares approach using the technique of singular value decomposition.

2.2. Estimation of EBF parameters

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 k-th class of the training set, $\mathcal{X}^{(k)}$, into $J^{(k)}$ disjoint clusters $\left\{\mathcal{X}_j^{(k)}\right\}_{j=1}^{J^{(k)}}$.[1] Thus, we estimate the function center $\vec{\mu}_j$ by the sample average $\hat{\vec{\mu}}_j$, i.e. $\vec{\mu}_j \approx \hat{\vec{\mu}}_j = \frac{1}{N_j}\sum_{\vec{x}\in\mathcal{X}_j}\vec{x}$ 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 $\mathcal{X}_j$, and $\|\cdot\|$ is the Euclidean norm. In the second step, the covariance matrices are approximated by the sample covariances, i.e.

$$\mathbf{\Sigma}_j \approx \hat{\mathbf{\Sigma}}_j = \frac{1}{N_j}\sum_{\vec{x}\in\mathcal{X}_j}(\vec{x} - \hat{\vec{\mu}}_j)(\vec{x} - \hat{\vec{\mu}}_j)^T. \tag{3}$$

Although it has been shown that EBF networks trained by the above two-step approach could give superior performance as compared to RBF networks [6], it may lead to undesirable results when the estimated mean $\hat{\vec{\mu}}_j$ differs significantly from the true mean $\vec{\mu}_j$. Consequently, the covariance matrix $\hat{\mathbf{\Sigma}}_j$ will no longer be accurate as an inaccurate mean vector has been used in (3).

To solve this problem, we need an iterative procedure by which the estimated means as well as the estimated covariance matrices move closer to the maximum likelihood estimates after each iteration. This idea leads to the EM algorithm [2] 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

$$\vec{\mu}_j^{\text{new}} = \frac{\sum_{\vec{x}\in\mathcal{X}} P^{\text{old}}(j|\vec{x})\vec{x}}{\sum_{\vec{x}\in\mathcal{X}} P^{\text{old}}(j|\vec{x})}, \tag{4}$$

$$\mathbf{\Sigma}_j^{\text{new}} = \frac{\sum_{\vec{x}\in\mathcal{X}} P^{\text{old}}(j|\vec{x})(\vec{x} - \vec{\mu}_j^{\text{new}})(\vec{x} - \vec{\mu}_j^{\text{new}})^T}{\sum_{\vec{x}\in\mathcal{X}} P^{\text{old}}(j|\vec{x})}, \text{ and} \tag{5}$$

$$P^{\text{new}}(j) = \frac{\sum_{\vec{x}\in\mathcal{X}} P^{\text{old}}(j|\vec{x})}{\sum_{r=1}^{J} N_r}, \tag{6}$$

respectively $\forall j = 1, \ldots, J$. In (4) to (6), $P^{\text{old}}(j|\vec{x})$ is the posterior probability of the jth cluster, which can be obtained by using the Bayes' theorem, yielding

$$P^{\text{old}}(j|\vec{x}) = \frac{p(\vec{x}|j)P^{\text{old}}(j)}{\sum_k p(\vec{x}|k)P^{\text{old}}(k)} \tag{7}$$

where

$$p(\vec{x}|j) = \frac{\exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_j^{\text{old}})^T \left(\mathbf{\Sigma}_j^{\text{old}}\right)^{-1}(\vec{x} - \vec{\mu}_j^{\text{old}})\right\}}{(2\pi)^{\frac{I}{2}}|\mathbf{\Sigma}_j^{\text{old}}|^{\frac{1}{2}}} \tag{8}$$

is the probability density function of the jth cluster. In case the covariance matrices are diagonal, (5) and (8) are modified to

$$\left(\sigma_{ji}^{\text{new}}\right)^2 = \frac{\sum_{\vec{x}\in\mathcal{X}} P^{\text{old}}(j|\vec{x})(x_i - \mu_{ji}^{\text{new}})^2}{\sum_{\vec{x}\in\mathcal{X}} P^{\text{old}}(j|\vec{x})} \quad \text{and} \tag{9}$$

$$p(\vec{x}|j) = \frac{1}{(2\pi)^{\frac{I}{2}}\prod_{i=1}^{I}\sigma_{ji}^{\text{old}}}\exp\left\{-\frac{1}{2}\sum_{i=1}^{I}\frac{(x_i - \mu_{ji}^{\text{old}})^2}{\left(\sigma_{ji}^{\text{old}}\right)^2}\right\}, \tag{10}$$

respectively.

3. APPLICATION TO SPEAKER VERIFICATION

3.1. Speech Corpus and Feature Extraction

A phonetically balanced, continuous speech corpus called TIMIT was used in the experiments. 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), anti-speaker set (38 speakers

[1] To simplify the notation, we drop the superscript (k) in the rest of this paper. Therefore, $\mathcal{X}$ denotes the training subset associated with one of the classes.

from region 1), pseudo-impostor 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. 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.

3.2. Enrollment

Each speaker in the speaker set was assigned a personalized network (RBF or EBF) modeling the characteristics of his or her own voice. Each network was trained to recognize the feature vectors derived from two classes (speaker class and anti-speaker class) derived respectively from the speaker set and the anti-speaker set. Thus, the networks comprise 12 inputs, various number of hidden nodes, and two outputs, with each output representing one class.

For each RBF network, the K-means algorithm was applied to the corresponding speaker and all anti-speakers separately (i.e. clustering within individual classes) to obtain the function centers. Next, the K-nearest neighbor algorithm with K set to 2 was applied to the resulting function centers to determine the function widths. Finally, the output weights were determined by the technique of singular value decomposition. For the EBF networks, the function centers and covariance matrices were determined by the EM algorithm ((4)–(10)) or sample covariance (3), and γ_m in $\phi_m(\vec{x})$ was determined by (2).

3.3. Verification

The network output $y_k(\vec{x})$ is an estimate of the *a posteriori* probability of the class C_k because an 1-of-K coding scheme has been used and the output units are linear [8].[2] As the ratio of training vectors between the speaker class and the anti-speaker class is about 1 to 38,[3] the network will favor the anti-speaker class during verification by always giving an output close to 1.0 for the anti-speaker class and close to 0.0 for the speaker class. This problem can be solved by scaling the outputs during verification so that the new average outputs are approximately equal to 0.5 for both classes. This can be achieved by dividing the output $y_k(\vec{x})$ by $2P(C_k)$, where $P(C_k)$ is the prior probability of class C_k. Specifically, we compute the scaled output $\tilde{y}_k(\vec{x}) = \frac{y_k(\vec{x})}{2P(C_k)}$ with $k = 1, 2$ so that $\frac{1}{N'}\sum_{x \in \mathcal{X}'} \tilde{y}_k(\vec{x}) \approx 0.5$, where N' is the number of training vectors 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.

For each verification session, we concatenated the test vectors from the utterances of a claimant to form a test sequence $\mathcal{T} = [\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_T]$, where T is the number of patterns in $\mathcal{T}$. The sequence was then divided into a number of overlapping segments containing 200 consecutive vectors (2.8 seconds of speech). For each segment $\mathcal{T}_s$ (with segment length T_s), the scaled average outputs

$$z_k = \frac{1}{T_s} \sum_{\vec{x} \in \mathcal{T}_s} \frac{\exp\{\tilde{y}_k(\vec{x})\}}{\sum_{r=1}^{2} \exp\{\tilde{y}_r(\vec{x})\}} \quad k = 1, 2 \qquad (11)$$

corresponding to the speaker and anti-speaker classes were computed. Note that we have made use of the softmax function inside the summation of (11). 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:

$$\text{If } z_1 - z_2 \begin{cases} > \zeta & \text{accept the claimant} \\ \leq \zeta & \text{reject the claimant} \end{cases} \qquad (12)$$

where $\zeta \in [-1, 1]$ is a threshold controlling the false rejection rate (FRR) and the false acceptance rate (FAR). FRR is the rate of falsely rejecting a true speaker, while FAR measures the rate of incorrectly accepting impostors. A verification decision was made for every segment. In other words, after each verification decision, a window covering 200 consecutive cepstral 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.

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 (11)), 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 'fame0' (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 im-

[2] This is true provided that the smoothing parameters γ_m are sufficient large so that the networks do not overfit the training data.

[3] For each network, training vectors were derived from the corresponding speaker and 38 anti-speakers.

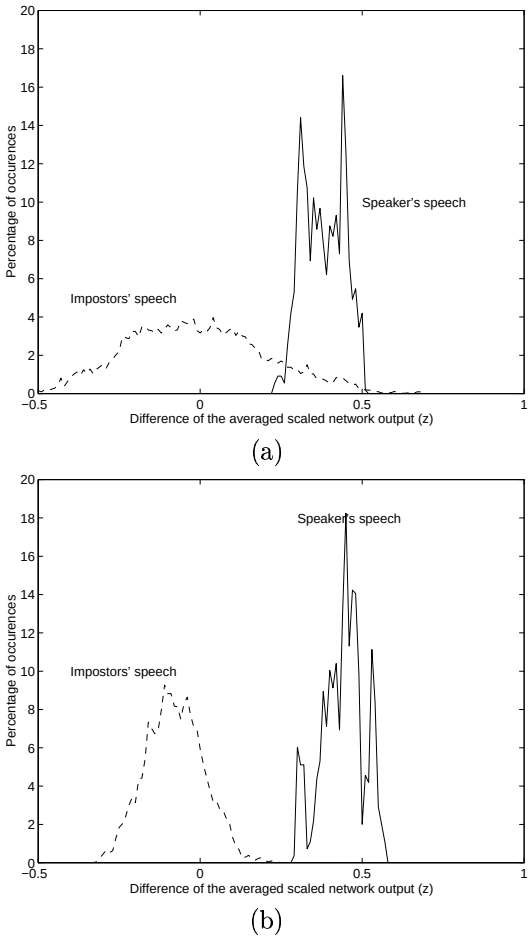

Figure 1: The distributions of z corresponding to (a) an RBF network and (b) an EBF network. Both networks contain 12 centers, 4 from the speaker and 8 from the anti-speakers.

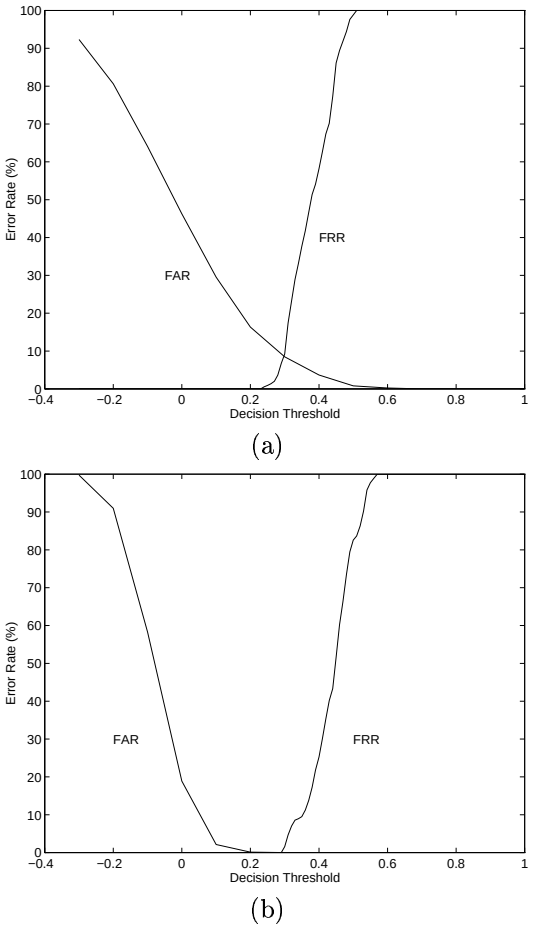

Figure 2: FAR and FRR versus the decision threshold of (a) an RBF network and (b) an EBF network. Both networks contain 12 centers, 4 from the speaker and 12 from the anti-speakers.

postors 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.

3.4. 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 above was applied. However, instead of using the speech of an unknown speaker, the feature vectors of pseudo-impostors in the pseudo-impostor 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 has been found, the false rejection rate corresponding to each speaker was obtained by presenting 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.

4. RESULTS AND DISCUSSION

We have tried various combinations of network type (RBF and EBF) and learning algorithms (K-means, K-nearest neighbors, sample covariance, and EM). Table 1 summarizes the verification experiments we have conducted.

Table 2 shows the false acceptance rates (FAR's), false rejection rates (FRR's), and equal error rates (ERR's) for different network types, network sizes, and learning algorithms. The equal error rates were obtained by adjusting the threshold until FAR is equal to FRR. All error rates in Table 2 were based on the average of 76 speakers in the speaker set.

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

Table 1: Abbreviations of experiment titles, network types, and algorithms used in estimating the basis function parameters.

The results of Table 2 demonstrate the superiority of the EBF networks over the RBF networks. In particular, Table 2 shows that the equal error rate of the smallest EBF network (*EEF* with 10 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 2 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 (3) may give poorer estimates of the true covariance matrices. Table 2 also allows us to compare the performance of networks with different numbers of speaker centers. The last four rows of Table 2 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 anti-centers 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 widths 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.

We also observed that the EM algorithm becomes numerically unstable when the number of function centers is large. This problem is particularly serious for full covariance matrices, where numerical overflow will sometimes occurs when the probability density function (8) is evaluated. This might be due to the fact that the number of parameters in the EBF networks increases rapidly with the network size. For networks with diagonal covariance matrices, the number of parameters is $\mathcal{N}_{\text{width}} + \mathcal{N}_{\text{weight}} + \mathcal{N}_{\text{center}} = MI + (M+1)K + MI$, where I, M, and K are the numbers of inputs, centers, and outputs, respectively. However, for EBF networks with full covariance matrices, the number of parameters is $\frac{1}{2}MI(I+1) + (M+1)K + MI$. If I is large (12 in this study), the training set may not be large enough for the EM algorithm to determine the parameters accurately. This could cause zero variances in the covariance matrices, resulting in numerical overflow.

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 3 lists the equal 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.

5. CONCLUSIONS

We proposed to apply the EM algorithm to estimate the parameters of elliptical basis function networks. We have evaluated and compared the performance of the EBF and RBF networks through a series of text-independent speaker verification experiments. The results confirmed the claim by Lin and Kung [5] that the use of elliptical basis functions with diagonal covariance matrices can lead to superior performance than the use of radial basis functions (with equal variances in all input dimensions). Furthermore, our results enable us to concur and further suggest that this idea can be extended to full covariance matrices, provided that care

Abbr.	Number of Centers per Network											
	10 (8+2)			12 (8+4)			16 (8+8)			24 (8+16)		
	FAR	FRR	**ERR**	FAR	FRR	**ERR**	FAR	FRR	**ERR**	FAR	FRR	**ERR**
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	**ERR**	FAR	FRR	**ERR**	FAR	FRR	**ERR**	FAR	FRR	**ERR**
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	18.95	**0.17**	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	3.35	7.54	**0.14**	4.95	0.75	**0.05**	8.32	0.00	**0.03**	7.39	0.00	**0.02**

Table 2: FAR's, FRR's, and ERR's (in %) for networks with various numbers of centers. Each network contains 2 to 16 centers contributed from the corresponding speaker and the rest are from the anti-speakers. For example, the network with 10 centers has 8 centers from the corresponding speaker and 2 from the anti-speakers, i.e. (8+2) centers.

Abbr.	**No. of centers**	**No. of Parameters**	**ERR (%)**
R	61 (12 + 49)	917	**3.85**
EC	10 (2 + 8)	922	**0.17**
EED	35 (7 + 28)	912	**0.40**
EEF	10 (2 + 8)	922	**0.14**

Table 3: Equal error rates (ERRs) for networks with comparable numbers of parameters. The second column lists the total number of centers per network, including speaker centers and anti-centers. For example, $(12+49)$ means that the networks have 12 speaker centers and 49 anti-centers. The ratio between speaker centers and anti-centers is approximately equal to 4.0.

is taken to ensure that sufficient training vectors are available to estimate the EBF parameters accurately.

6. REFERENCES

[1] J. Moody and C. J. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, 1:281–194, 1989.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of Royal Statistical Soc., Ser. B.*, 39(1):1–38, 1977.

[3] D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Trans. on Speech and Audio Processing*, 3(1):72–83, 1995.

[4] Hans. G. C. Tråvén. A neural network approach to statistical pattern classification by semiparametric estimation of probability density functions. *IEEE Tran. on Neural Networks*, 2(3):366–377, May 1991.

[5] S. H. Lin and S. Y. Kung. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. on Neural Networks, Special Issue on Biometric Identification*, 8(1):114–132, 1997.

[6] M. W. Mak, C. K. Li, and X. Li. Maximum likelihood estimation of elliptical basis function parameters with application to speaker verification. In *Proc. Int. Conf. on Signal Processing*, pages 1287–1290, 1998.

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

[8] M. D. Richard and R. P. Lippmann. Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation*, 3:461–483, 1991.