SPEAKER IDENTIFICATION USING ELLIPTICAL BASED NEURAL NETWORKS

Babi Kiran.V
Sandeep.S
Sanjay Premnadh.M
Siva Ramaiah.Y

Abstract—
An elliptical basis function (EBF) network is proposed in this study for the speaker identification. Though similar in structure, the EBF network differs from the well-known radial basis function (RBF) network by incorporating full covariance matrices and uses the expectation-maximization (EM) algorithm to estimate the basis functions. Experimental evaluations based on 100 speakers 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

Keywords- RadialBasisFunctions, EM Algorithm, Hyperelipsoidal, Speaker Verification

I. Introduction

Radial Based is Function (RBF) networks have successfully been applied to a wide range of pattern recognition problems. 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 control ling the spread of the corresponding RBF unit. As a result the RBF units are hyper-spherical, 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 bas is function can still be considered to have independent components. This paper, therefore will introduce elliptical bas is function (EBF) networks with full covariance matrices in an attempt to enhance the classification capability of conventional RBF networks

II. EBF versus RBF Networks

A. *Architecture of EBF Networks*

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

$y_{k(\overline{x_p})} = w_{k0} + \sum_{j=1}^{M} w_{kj}(\overline{x_p})$ p=1,.,N and k=1,.,K (1)

Where

$\boldsymbol{\phi}_j (\overline{x_p}) = \{- \frac{1}{2\gamma_j}(\overline{x_p} - \overline{\mu_j})^T \Sigma_j^{-1}(\overrightarrow{x_p} - \overrightarrow{\mu_j})\}$ j=1,…, M (2)

In (1) and (2) $\overrightarrow{x_p}$ is the pth input vector , $\overrightarrow{\mu_j}$ and Σ_j are the mean vector and covariance matrix of the jth basis function respectively,w_{k0} is a bias term , and $\boldsymbol{\gamma}_j$ is a smoothing parameter controlling the spread of the jth basis function. In this work $\boldsymbol{\gamma}_j$ was determined heuristically by

$$\gamma_j = \frac{3}{5} \sum_{k=1}^{5} ||\overrightarrow{\mu_k} - \overrightarrow{\mu_j}||$$

Where $\overrightarrow{\mu_k}$ denotes the k-th nearest neighbor of $\overrightarrow{\mu_j}$ in the Euclidean sense. Note that this method is similar to the K-nearest neighbour heuristic commonly used in determining the function widths of RBF networks. We have empirically found that using five nearest centres 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 centres is less than 5, the number of nearest centres 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 x K matrix, $\mathbf{\Phi}$ is an N x (M+ 1) matrix, and $\mathbf{W}$ is an (M+ 1) x K matrix. The weight matrix $\mathbf{W}$ is the least squares solution of the matrix equation $\mathbf{\Phi}\ \mathbf{W} = \mathbf{D}$, where D is an N x K target matrix containing the desired output vectors in its rows. As Φ is not a square matrix, one reliable way to $\mathbf{W}$ is to use the technique of singular value decomposition

B. Estimation of EBF Parameters

B.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 k-th c lass of the training set, $\chi^{(k)}$, into $\mathbf{J}^{(k)}$ disjoint clusters. $\{\chi_j^{(k)}\}_{j=1}^{J^{(k)}}$ Therefore, we estimate the function centre $\overrightarrow{\mu_j}$ by the sample average $\ddot{\overrightarrow{\mu_j}}$

$$\overrightarrow{\mu_j} \approx \widehat{\overrightarrow{\mu_j}} = \frac{1}{N_j} \sum_{\vec{x} \in \chi_j} \vec{x} \qquad (4)$$

Where $\vec{x} \in \chi_j$ if $\|\vec{x} - \ddot{\overrightarrow{\mu_j}}\| < \|\vec{x} - \ddot{\overrightarrow{\mu_k}}\| \forall j \neq k$, N_j is the number of samples in the cluster χ_j and $\|.\|$ is the Euclidean norm. In the second step, the covariance matrices are approximated by the sample covariance

$$\Sigma_j \approx \hat{\Sigma}_j = \frac{1}{N_j} \sum_{\vec{x} \in \chi_j} (\vec{x} - \ddot{\overrightarrow{\mu_j}})(\vec{x} - \ddot{\overrightarrow{\mu_j}})^T \qquad (5)$$

Although it has been shown that EBF networks trained in the above two-step approach may give performance superior to RBF networks, they may also cause undesirable results when the estimate $\ddot{\overrightarrow{\mu_j}}$ differs significantly from the true mean $\overrightarrow{\mu_j}$.Consequently, the covariance matrix $\hat{\Sigma}_j$. 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 estimate after each iteration. This idea points to the EM algorithm 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

$$\vec{\mu}_j^{new} = \frac{\sum_{\vec{x} \in X} p^{old}(j/\vec{x})\vec{x}}{\sum_{\vec{x} \in X} p^{old}(j/\vec{x})} \qquad (6)$$

$$\Sigma_j^{new} = \frac{\sum_{\vec{x} \in X} p^{old}(j/\vec{x})(\vec{x}-\vec{\mu}_j^{new})(\vec{x}-\vec{\mu}_j^{new})^T}{\sum_{\vec{x} \in X} p^{old}(j/\vec{x})} \; and \; (7)$$

$$P^{new}(j) = \frac{\sum_{\vec{x} \in X} p^{old}(j/\vec{x})}{\sum_{r=1}^{J} N_r} \qquad (8)$$

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

$$p^{old}\left(j/\vec{x}\right) = \frac{p(\vec{x}/j)p^{old}(j)}{\sum_k p(\vec{x}/k)p^{old}(k)} \qquad (9)$$

where

$$P(\vec{x}/j) = \frac{1}{(2\pi)^{\frac{1}{2}}|\Sigma_j^{old}|^{\frac{1}{2}}} \exp\{-\tfrac{1}{2}(\vec{x}-\vec{\mu}_j^{old})^T (\Sigma_j^{old})^{-1}(\vec{x}-\vec{\mu}_j^{old})\}$$

(10)

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

$$(\sigma_{ji}^{new})^2 = \frac{\sum_{\vec{x} \in X} p^{old}(j/\vec{x})(\vec{x}-\vec{\mu}_{ji}^{new})^2}{\sum_{\vec{x} \in X} p^{old}(j/\vec{x})} \; i = 1, \dots, I \qquad (11)$$

and

$$P(\vec{x}/j) = \frac{1}{(2\pi)^{\frac{1}{2}}\prod_{i=1}^{I}\sigma_{ji}^{old}} \, exp\,\{-\tfrac{1}{2}\sum_{i=1}^{I}\frac{(\vec{x}-\vec{\mu}_{ji}^{old})^2}{(\sigma_{ji}^{old})^2}\} \qquad (12)$$

respectively. Note that if P^{old} (j/$\vec{x}$) is equal to 1.0 for all $\vec{x} \in \chi_j$ 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 is a special case 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; 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. SPEAKER IDENTIFICATION

A.Enrollment

Each speaker in the speaker set was assigned a personalized network (RBF or EBF) modelling 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 ant i-speaker class. The former was derived from the speaker set while the latter from the ant i-speaker set. Therefore, each network was composed of 12 inputs, varied numbers of hidden nodes, and two outputs, with each output representing one class

The enrollment procedure consists of five steps. These are described below

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

Step 2: Apply the K-means algorithm to the cepstral vectors of all anti-speakers in the anti-speaker set to obtain a pool of function centres. These centres are referred to as the anti-centres

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 Step1 as $\vec{\mu}_j$.Similarly, (5) is applied to the cepstral vectors of the anti-speakers to obtain the widths corresponding to the ant i-centers. Then, go to Step 4

(b) If the network is an RBF one, apply the K-nearest neighbours algorithm (with K = 2) to the ant i-centers to obtain the function widths corresponding to the anti-centers. 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 bas is 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 3 (b) 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 $\vec{\mu}_j^{old}$ 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 **W**

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

Note that the above clustering procedure (Steps 1 to 3) was applied to the speaker class and the anti-speaker 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 anti-centers" which only need to be determined once. In the conventional approach, however, the anti-centers 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 ant i-speakers (38 in this study) to improve the capability of the networks in mode ling impostors speech

C. Veriication

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 $(C_k|\vec{x})$ 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 towards those classes with a larger proportion of patterns. For instance, in a two-class problem (i.e. = 2) where the prior probability of one classis 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 anti-speaker class is about 1 to 38.This is because each network uses one speaker from the speaker set and 38 speakers from the anti-speaker set for training. The network will favour the anti-speaker c lass during verification by always giving outputs which are close to one for the anti-speaker 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. 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 $\frac{1}{2P(C_k)}$. Specifically, we computed the scaled output

$$\overline{y_k}(\vec{x}) = \frac{1}{2} \cdot \frac{y_k(\vec{x})}{P(C_k)} \qquad k=1, 2 \qquad (13)$$

so that $\frac{1}{N'}\sum_{x\in\chi'}\overline{y_k}(\vec{x}) \approx 0.5$ where N' denotes the number of patterns in the training set χ' . 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 T = [$\vec{x_1}$…...$\vec{x_T}$] corresponding to an utterance spoken by an unknown speaker was fed into the network. Then we computed the scaled average outputs

$$z_k = \frac{1}{2} \cdot \sum_{\vec{x}\in T} \frac{\exp\{\overline{y_k}(\vec{x})\}}{\exp\{\overline{y_1}(\vec{x})\}+\exp\{\overline{y_2}(\vec{x})\}} \qquad k=1,2 \quad (14)$$

Corresponding to the speaker and ant i-speaker 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 $\overline{y_k}$ from dominating the average outputs .Verification decisions were based on the criterion:

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

Where $\zeta \in [-1,1]$ is a threshold control ling 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 T' = [$\vec{x_1}$,$\vec{x_2}$,….$\vec{x_T}$].The sequence was then divided into a number of over lapping segments containing 200 consecutive vectors (2.8 seconds 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 `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 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 over lapped). 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 Sect ion III-C 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 value 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 work; the predefined FAR was set to 2%.

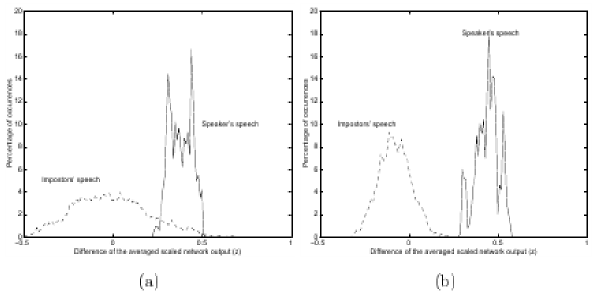

Fig.1The distributions of z corresponding to (a) an RBF network and (b) an EBF network. Both networks contain 12 centers, 4 from the speaker and 8 from the anti-speaker.

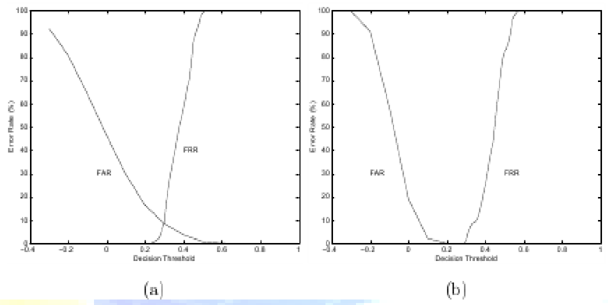

Fig.2.FAR and FRR versus the decision threshold of (a) an RBF network and (b) an EBF network. Both networks contain 12 centers, 4 from the speaker and 12 from the anti-speakers.

Once the threshold value 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 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 summarize the verification ion experiments we have conducted.

Table II summarizes the false acceptance rates (FAR's), false reject ion rates (FRR's), and equal error rates (EER's) 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 Tab le II 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 I

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

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 II

FAR's, FRR's, and EER'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.

The results of Tab le 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 1 0 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

IV. Conclusion

In this paper, we proposed to apply the EM algorithm to estimate the bas is funct ion parameters of elliptical basis funct ion networks. The proposed learning scheme enables the maximum likelihood estimates of the EBF parameters to be found, resulting in 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. The conclusion can be drawn from the results of these experiments. Firstly, we have found that for the same number of funct ion centers, EBF networks with full covariance matrices trained with the EM algorithm outperform the ones whose basis function parameters are estimated by sample covariance Secondly, RBF networks are found to be the poorest performers in terms of verification accuracy. Finally, this study has shown that when the numbers of free parameters are comparable, EBF networks with full covariance matrices achieve the lowest equal error rate

V. References

[1] S.Renals. Radial bas is funct ion for speech pattern Classification. Electronic Letters, 25 (7):437{439, 1989

[2] M.W. Mak, W. G. Allen, and G. G. Sexton. Speaker identification using multilayer perceptrons and radial basis funct ion networks. Neuro computing, 6: 99{1 17, 1994

[3] M.W. Mak, W. G. Allen, and G. G. Sexton. Speaker identification using multilayer perceptrons and radial basis funct ion networks. Neuro computing, 6: 99{1 17, 1994

[4] J. Moody and C. J. Darken. Fast learning in networks of locally tuned processing unit s .Neural Computation, 1:28 1{194, 1989

[5] L. Xu. RBF nets, mixture experts, and Bayesian Ying-Yang learning. Neuro computing, 19:223{257, 1 998

[6] A. R. Webb. Functional approximation by feed forward networks: a least -squares approach to generalization IEEE Transaction Neural Network s, 5 (3):363{371, 1994.

[7] H. Schioler and U. Hartmann. Mapping neural network derived from the Parzen window estimator. Neural Networks, 5 (6):903{909, 1992

8] D. F. Specht. Probabilistic neural networks. Neural Networks, 3: 109{1 18, 1990.

[9] T. Poggio and F. Girosi. Networks for approximation and learning. Proceedings of the IEEE, 78 (9): 148 1{1497, 1990