SPEAKER VERIFICATION WITH A PRIORI THRESHOLD DETERMINATION USING KERNEL-BASED PROBABILISTIC NEURAL NETWORKS

Kwok-Kwong Yiu and Man-Wai Mak

Center for Multimedia Signal Processing
Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University, China

Sun-Yuan Kung[♯]

Dept. of Electrical Engineering
Princeton University
USA

ABSTRACT

This paper compares kernel-based probabilistic neural networks for speaker verification. Experimental evaluations based on 138 speakers of the YOHO corpus using probabilistic decision-based neural networks (PDBNNs), Gaussian mixture models (GMMs) and elliptical basis function networks (EBFNs) as speaker models were conducted. The original PDBNN training algorithm was also modified to make PDBNNs appropriate for speaker verification. Results show that the equal error rate obtained by PDBNNs and GMMs is about half of that of EBFNs (1.19% vs. 2.73%), suggesting that GMM- and PDBNN-based speaker models outperform the EBFN one. This work also finds that the globally supervised learning of PDBNNs is able to find a set of decision thresholds that reduce the variation in FAR, whereas the ad hoc approach used by the EBFNs and GMMs is not able to do so. This property makes the performance of PDBNN-based systems more predictable.

1. INTRODUCTION

In speaker verification systems, each registered speaker is assigned a speaker-dependent model characterizing his or her own voice. Typically, each of these models is trained to estimate the likelihood of the corresponding speaker given an utterance. Gaussian mixture models (GMMs) [1] and elliptical basis function networks (EBFNs) [2] have been widely used as speaker models because of their capability to model arbitrary density functions. However, GMMs and EBFNs have limitations as they do not provide a proper mechanism for setting decision thresholds, making the verification systems vulnerable to impostor attacks. Therefore, a more advanced speaker model is needed.

Probabilistic decision-based neural networks (PDBNNs), proposed by Lin et al. [3], can be considered as a special form of GMMs with trainable decision thresholds. PDBNNs were used to implement a hierarchical face recognition system in [3] with excellent results (97.75% recognition, 2.25% false rejection, 0% misclassification and 0% false acceptance). The characteristics of PDBNNs' decision boundaries have been investigated in our previous study [4], where the strengths of PDBNNs are highlighted by comparing the recognition accuracy and decision boundaries of PDBNNs against those of GMMs. We have also demonstrated in [4] that the thresholding mechanism of PDBNNs is very effective in detecting data not belonging to any known classes. In light of this finding, this paper applies PDBNNs to speaker verification in an attempt to improve the robustness of speaker verification systems against intruder attacks.

2. PROBABILISTIC DECISION-BASED NEURAL NETWORKS

Probabilistic decision-based neural networks (PDBNNs) are a probabilistic variant of their predecessor, DBNNs [5], for robust pattern classification. PDBNNs employ a modular network structure. In other words, a PDBNN is composed of a number of small sub-networks, with each subnetwork representing one class. Each class follows a probabilistic constraint, and the likelihood function $p(x(t)|\omega_i)$ for each class ω_i is a mixture of Gaussian distributions. The subnet discriminant functions of a PDBNN are designed to model some log-likelihood functions of the form

$$\begin{aligned}\phi(x(t), \mathbf{w}_i) &= \log p(x(t)|\omega_i) \\ &= \log\left[\sum_{r=1}^{R} P(\Theta_{r|i}|\omega_i) p(x(t)|\omega_i, \Theta_{r|i})\right]\end{aligned} \quad (1)$$

where $\mathbf{w}_i \equiv \left\{\mu_{r|i}, \Sigma_{r|i}, P(\Theta_{r|i}|\omega_i), T_i\right\}$, $\Theta_{r|i}$ represents the parameters of the rth mixture component, R is the total number of mixture components, $p(x(t)|\omega_i, \Theta_{r|i})$ is the probability density function of the rth component and $P(\Theta_{r|i}|\omega_i)$ is the prior probability (also called mixture coefficients) of the rth component and T_i is the decision

This project was supported by the Hong Kong Polytechnic University Grant No. G-W076 and RGC Project No. PolyU 5129/01E. ♯S.Y. Kung was also a Distinguished Chair Professor of The Hong Kong Polytechnic University.

threshold of the i-th subnet. Typically, $p(x(t)|\omega_i, \Theta_{r|i})$ is a Gaussian distribution with mean $\mu_{r|i}$ and covariance $\Sigma_{r|i}$.

Learning in PDBNNs is divided into two phases: locally unsupervised (LU) and globally supervised (GS). In the LU learning phase, each subnet is trained independently, and no mutual information across the classes is used. Specifically, PDBNNs adopt the expectation-maximization (EM) algorithm [6] to maximize the log-likelihood function

$$l(\mathbf{w}_i; \mathcal{X}_i) \quad = \quad \sum_{t=1}^{N} \phi(x(t), \mathbf{w}_i) \qquad (2)$$

with respect to the parameters $\mu_{r|i}$, $\Sigma_{r|i}$, and $P(\Theta_{r|i}|\omega_i)$, where $\mathcal{X}_i = \{x(t); t = 1, 2, \ldots, N\}$ denotes the set of N independent and identically distributed training patterns from class ω_i.

In the globally supervised (GS) training phase, target values are utilized to fine-tune the decision boundaries. The network weights will be updated whenever misclassification occurs. Specifically, *reinforced learning* is applied to the subnet corresponding to the correct class so that the weight vector is updated in a direction parallel to the gradient of $\phi(x, \mathbf{w}_i)$, whereas *anti-reinforced learning* is applied to the (unduly) winning subnet to move the weight vector along the opposite direction:

$$\begin{aligned} \mathbf{w}_i^{(j+1)} &= \mathbf{w}_i^{(j)} + \eta \nabla \phi(x, \mathbf{w}_i) \quad \text{(reinforced)} \\ \mathbf{w}_i^{(j+1)} &= \mathbf{w}_i^{(j)} - \eta \nabla \phi(x, \mathbf{w}_i) \quad \text{(anti-reinforced)} \end{aligned}$$

This has the effect of increasing the chance of classifying the same pattern correctly in the future.

3. APPLICATIONS TO SPEAKER VERIFICATION

3.1. Enrollment Procedures

Each registered speaker was assigned a personalized network (GMM, EBFN or PDBNN) which was trained to recognize the speech derived from two classes—speaker class and anti-speaker class. To this end, two groups of kernel functions (one representing the speaker himself/herself while the other representing the speakers in the anti-speaker class) were assigned to each network. We denote the group corresponding to the speaker class as the speaker kernels and the one corresponding to the anti-speaker class as the anti-speaker kernels. For each registered speaker, a unique anti-speaker set containing a predefined number of anti-speakers was created. This set was used to create the anti-speaker kernels. The anti-speaker kernels enable us to incorporate the idea of scoring normalization [7] in the training procedures, which enhances the networks' capability in discriminating the true speakers from the impostors.

Each of the GMMs and PDBNNs is composed of 12 inputs (12th-order cepstral coefficients were used as features), a pre-defined number of kernels, and one output. On the other hand, the EBFNs contain 12 inputs, a pre-defined number of kernels, and 2 outputs with each output representing one class (speaker class or anti-speaker class). All of the covariance matrices in the kernels are diagonal.

We applied the K-means algorithm to initialize the positions of the speaker kernels. Then, the kernels' covariance matrices were initialized by the K-nearest neighbors algorithm ($K = 2$). In other words, all off-diagonal elements were zero and the diagonal elements (being equal) of each matrix were initialized to the average Euclidean distance between the corresponding center and its K-nearest centers. The EM algorithm was subsequently used to fine-tune the mean vectors, covariance matrices, and mixture coefficients. The same procedure was also applied to determine the mean vectors and covariance matrices of the anti-speaker kernels, using the speech data derived from the anti-speaker set.

The enrollment process for constructing a PDBNN-based speaker model involves two phases: locally unsupervised (LU) training and globally supervised (GS) training. The LU training phase is identical to the GMM training described above. In the GS training phase, the speaker's enrollment utterances and the utterances from all enrollment sessions of the anti-speakers were used to determine a decision threshold (see Section 3.3 below).

For the EBFN-based speaker models, the speaker kernels and anti-speaker kernels obtained from the GMM training described above were combined to form a hidden layer. Finally, singular value decomposition was applied to determine the output weights. Details of the enrollment procedure for EBFNs can be found in [2].

3.2. Verification Procedures

Verification was performed using each speaker in the YOHO corpus as a claimant, with 64 impostors being randomly selected from the remaining speakers (excluding the anti-speakers and the claimant) and rotating through all the speakers. For each claimant, the feature vectors of the claimant's utterances from his/her 10 verification sessions in YOHO were concatenated to form a claimant sequence. Likewise, the feature vectors of the impostor's utterances were concatenated to form an impostor sequence.

For PDBNNs and GMMs, the following steps were performed during verification. The feature vectors from the claimant's speech $\mathcal{T}^c = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_{T_c}\}$ were divided into a number of overlapping segments containing $T (< T_c)$ consecutive vectors as shown below.

$$\overbrace{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4, \vec{x}_5, \vec{x}_6, \cdots\cdots, \vec{x}_T}^{\text{1st segment, } \mathcal{T}_1}, \vec{x}_{T+1}, \vec{x}_{T+2}, \cdots, \vec{x}_{T_c}$$

$$\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4, \vec{x}_5, \overbrace{\vec{x}_6, \cdots\cdots, \vec{x}_{T+5}}^{\text{2nd segment, } \mathcal{T}_2}, \vec{x}_{T+6}, \cdots, \vec{x}_{T_c}$$

For the t-th segment ($\mathcal{T}_t$), the average normalized log-likelihood

$$z_t = \frac{1}{T} \sum_{\vec{x} \in \mathcal{T}_t} \{\phi_S(\vec{x}) - \phi_A(\vec{x})\} \tag{3}$$

of the PDBNN-based and GMM-based speaker models was computed, where $\phi_S(\vec{x})$ and $\phi_A(\vec{x})$ represent the log-likelihood function (Eqn. 2) of the speaker and anti-speakers respectively. Verification decisions were based on the following criterion:

$$\text{If } z_t \begin{cases} > \zeta & \text{accept the claimant} \\ \leq \zeta & \text{reject the claimant} \end{cases} \tag{4}$$

where ζ is a speaker-dependent decision threshold (see Section 3.3 below for the procedure of determining ζ). A verification decision was made for each segment, with the error rate (either false acceptance or false rejection) being the proportion of incorrect verification decisions to the total number of decisions. In this work, T in Eqn. (3) was set to 500 (i.e., 7 seconds of speech), and each segment was separated by five vector positions. More specifically, the t-th segment contains the vectors

$$\mathcal{T}_t = \{\vec{x}_{5(t-1)+1}, \vec{x}_{5(t-1)+2}, \ldots, \vec{x}_{5(t-1)+500}\}$$

where $5(t-1) + 500 < T_c$. Note that dividing the vector sequence into a number of segments has also been successfully used in [1], [2] for increasing the number of decisions.

For the EBFN-based speaker models, verification decisions were based on the difference between the scaled network outputs [2]. Details of the verification procedure can be found in [2].

In this work, equal error rate (EER)—false acceptance rate being equal to false rejection rate—was used as a performance index to compare the verification performance among different speaker models. As the speaker models remain fixed once they have been trained, EER can be used to compare the models' ability in discriminating the speaker features from the impostor features.

3.3. Determination of Decision Thresholds

The procedures for determining the decision thresholds of PDBNNs, GMMs and EBFNs are different. For the GMM-based and EBFN-based speaker models, the utterances from all enrollment sessions of 16 randomly selected anti-speakers were used for threshold determination. Specifically, these utterances were concatenated and the procedure described in Section 3.2 was applied. The threshold ζ was adjusted until the false acceptance rate (FAR) fell below a pre-defined level. In this work, we set this level to 0.5%. The reason behind using anti-speakers' utterances rather than speaker's utterances is that it is much easier to collect the speech of a large number of anti-speakers. Hence, the thresholds obtained are more reliable than those that would be obtained from speaker's speech. In addition, using a pre-defined FAR to determine the decision thresholds enables us to predict the robustness of the verification system against impostor attacks [8].

To adopt PDBNNs to speaker verification and to determine the decision threshold of PDBNN-based speaker models, three modifications on the PDBNN's training algorithm have been made. First, the original PDBNNs use one threshold per network. However, in our case, for each speaker we use one network to model the speaker class and another one to model the anti-speaker class. To make PDBNNs applicable to speaker verification, we modified the likelihood computation such that only one threshold is required. Specifically, instead of comparing the subnet's log-likelihood against its corresponding threshold as in the original PDBNNs, we compared a normalized score against a single decision threshold as in Eqns. (3) and (4).

In the second modification, we changed the frequency at which the threshold is updated. The original PDBNN adopts the batch-mode supervised learning. Our speaker verification procedure, however, adopts a segmental type of learning (see Section 3.2). Specifically, we modified the GS training to work on a segmental mode as follows. Let $\mathcal{T}_n$ be the n-th segment extracted from speaker's speech patterns $\mathcal{X}_S$ or from anti-speakers' speech patterns $\mathcal{X}_A$, the normalized segmental score is computed by evaluating

$$\begin{aligned} S(\mathcal{T}_n) &= S_S(\mathcal{T}_n) - S_A(\mathcal{T}_n) \\ &= \frac{1}{T} \sum_{\vec{x} \in \mathcal{T}_n} \phi_S(\vec{x}) - \frac{1}{T} \sum_{\vec{x} \in \mathcal{T}_n} \phi_A(\vec{x}) \\ &= \frac{1}{T} \sum_{\vec{x} \in \mathcal{T}_n} \{\phi_S(\vec{x}) - \phi_A(\vec{x})\} \end{aligned}$$

where $\phi_S(\vec{x})$ and $\phi_A(\vec{x})$ denotes the log-likelihood (Eqn. (1)) of speaker's speech and impostors' speech respectively. For each segment, a verification decision was made according to the criterion:

$$\text{If } S(\mathcal{T}_n) \begin{cases} > \zeta_{n-1}^{(j)} & \text{accept the claimant} \\ \leq \zeta_{n-1}^{(j)} & \text{reject the claimant} \end{cases}$$

where $\zeta_{n-1}^{(j)}$ is the decision threshold of the PDBNN-based speaker model after learning from segment $\mathcal{T}_{n-1}$ at epoch j. We adjusted $\zeta_{n-1}^{(j)}$ whenever misclassification occurs. Specifically, we update $\zeta_{n-1}^{(j)}$ according to

$$\zeta_n^{(j)} = \begin{cases} \zeta_{n-1}^{(j)} - \eta_r l'(\zeta_{n-1}^{(j)} - S(\mathcal{T}_n)) \\ \quad \text{if } \mathcal{T}_n \in \mathcal{X}_S \quad \text{and} \quad S(\mathcal{T}_n) < \zeta_{n-1}^{(j)} \\ \zeta_{n-1}^{(j)} + \eta_a l'(S(\mathcal{T}_n) - \zeta_{n-1}^{(j)}) \\ \quad \text{if } \mathcal{T}_n \in \mathcal{X}_A \quad \text{and} \quad S(\mathcal{T}_n) \geq \zeta_{n-1}^{(j)} \end{cases}$$

where η_r and η_a are respectively the reinforced and anti-reinforced learning rates (more on next paragraph), $l(d) = \frac{1}{1+e^{-d}}$ is a penalty function, and $l'(d)$ is the derivative of $l(\cdot)$.

In the third modification, we introduced a new method to compute the learning rates. In the original PDBNNs, the learning rates for optimizing the thresholds are identical for both reinforced and anti-reinforced learning. However, in some situations, there may be many false acceptances and only a few false rejections (or vice versa), which means that anti-reinforced learning will occur more frequent than reinforced learning (or vice versa). In order to reduce the unbalance in the learning frequency, we make the reinforced (anti-reinforced) learning rate to be proportional to the number of false acceptance (rejections), i.e.,

$$\eta_r = \frac{N_{FA}^{(j-1)}}{N_{FA}^{(j-1)} + N_{FR}^{(j-1)}}\eta$$

$$\eta_a = \frac{N_{FR}^{(j-1)}}{N_{FA}^{(j-1)} + N_{FR}^{(j-1)}}\eta$$

where $N_{FR}^{(j-1)}$ and $N_{FA}^{(j-1)}$ represent respectively the total number of false rejections and false acceptances at epoch $j-1$ and η is a positive learning parameter. As a result, a frequent learner will have a smaller learning rate while a non-frequent learner will have a higher learning rate. This arrangement can prevent the reinforced learning or the anti-reinforced learning from dominating the learning process.

In this work, we only modified the decision thresholds in the globally supervised training with the mean vectors $\mu_{r|i}$ and covariance matrices $\Sigma_{r|i}$ remain unchanged. This is because we want to maintain the maximum likelihood nature of the models.

3.4. Speaker Verification Experiments

In this section, experimental evaluations on closed-set text-independent speaker verification based on all speakers (108 male, 30 female) in the YOHO corpus [9] are presented. We used 40 speaker kernels, 160 anti-speaker kernels, and 16 anti-speakers for creating each speaker model. The aim is to evaluate the robustness of different pattern classifiers (speaker models) for speaker verification. To demonstrate the robustness of different classifiers, speech from the enrollment sessions of the YOHO corpus was used for training while the speech from the verification sessions was used for testing. Verification was performed using each speaker in the corpus as a claimant, with 64 impostors being randomly selected from the remaining speakers (excluding the anti-speakers) and rotating through all speakers.

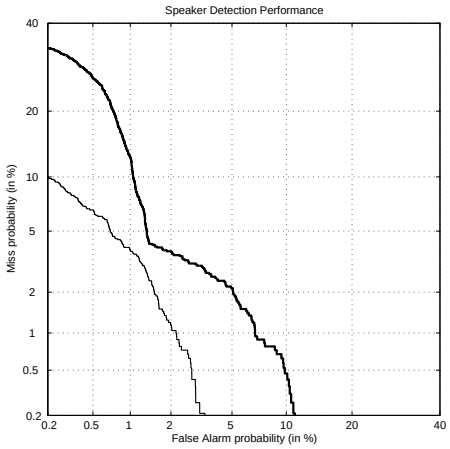


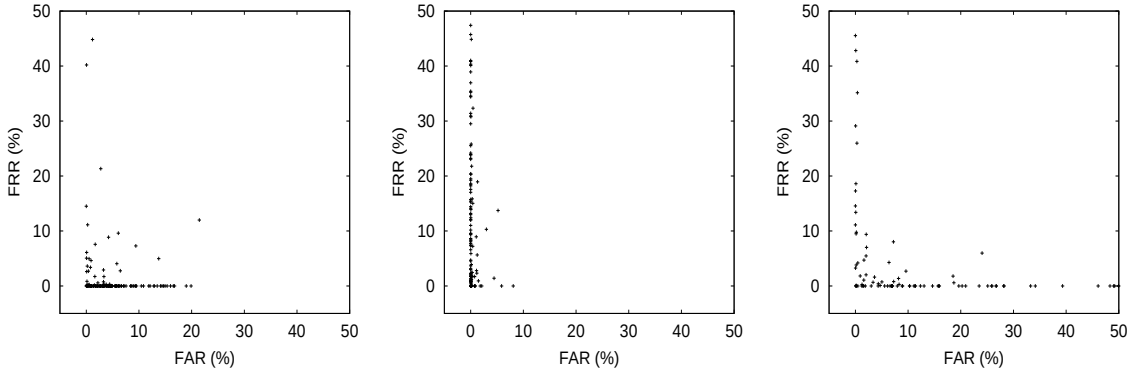
Figure 1: DET curves for speaker 277. Thick curve: EBFN-based speaker model. Thin curve: GMM-based and PDBNN-based speaker models. Note that the DET curves for the GMM and the PDBNN are identical since the GS training only updates the threshold of the PDBNN.

Speaker Model	FAR (%)	FRR (%)	EER (%)
GMMs	4.91	1.68	1.19
EBFs	33.51	4.93	2.73
PDBNNs	0.35	16.17	1.19

Table 1: Average error rates obtained by the GMMs, EBFNs and PDBNNs. The pre-defined FAR for GMMs and EBFNs was set to 0.5%.

Table 1 summarizes the average FAR, FRR, and EER obtained by the PDBNN-, GMM- and EBFN-based speaker models. All results are based on the average of 138 speakers in the YOHO corpus. The results, in particular the EER, demonstrate the superiority of the GMMs and PDBNNs over the EBFNs. The EER of GMMs and PDBNNs are the same since their kernel parameters are identical.

Table 1 also demonstrates the superiority of the threshold determination procedure of PDBNNs. In particular, Table 1 shows that the globally supervised learning of PDBNNs can make the average FAR very small during verification, whereas the ad hoc approach used by the EBFNs and GMMs is not able to do so. Recall from our previous discussion that the pre-defined FAR was set to 0.5%. The average FAR of EBFNs and GMMs are, however, very different from this value. This suggests that it may be difficult for us to predict the performance of the EBFNs and GMMs



(a) (b) (c)

Figure 2: FRRs versus FARs (during verification) of 138 speakers using (a) GMMs, (b) PDBNNs and (c) EBFNs as speaker models.

in detecting the impostor attacks.

Figure 1 shows the DET curves [10] for speaker 277 using different types of speaker models. In the DET plots, we use a nonlinear scale for both axes so that systems producing Gaussian distributed scores will be represented by straight lines. This property helps spread out the receiver operating characteristics (ROCs), making comparison of well-performed systems much easier. Note that the DET curves for GMM and PDBNN are identical in this experiment because the globally supervised training updates the thresholds of PDBNNs only. It is evident from Figure 1 that the GMM- and PDBNN- speaker models outperform the EBFN one.

Figure 2 depicts the FAR and FRR of individual speakers in the GMM-, EBFN- and PDBNN-based speaker verification systems. Evidently, most of the speakers in the PDBNN-based system exhibit a low FAR. On the other hand, the GMMs and EBFNs exhibit a much large variation in FAR. We conjecture that the globally supervised learning in PDBNNs is able to find decision thresholds that minimize the variation in FAR.

4. CONCLUSIONS

This paper addresses the problem of building speaker verification systems using kernel-based probabilistic neural networks such as GMMs, EBFNs and PDBNNs. The modelling capability of these pattern classifiers are compared. Experimental results indicated that GMM- and PDBNN-based speaker models outperform the EBFN ones. This work also finds that the globally supervised learning of PDBNNs can reduce the FARs and maintain their variation to a low level.

5. REFERENCES

[1] 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.

[2] M.W. Mak and S.Y. Kung. Estimation of elliptical basis function parameters by the EM algorithms with application to speaker verification. *IEEE Trans. on Neural Networks*, 11(4):961–969, 2000.

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

[4] K. K. Yiu, M. W. Mak, and C. K. Li. Gaussian mixture models and probabilistic decision-based neural networks for pattern classification: A comparative study. *Neural Computing & Applications*, 8:235–245, 1999.

[5] S. Y. Kung. *Digital Neural Networks*. Prentice Hall, New Jersey, 1993.

[6] 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.

[7] C. S. Liu, H. C. Wang, and C. H. Lee. Speaker verification using normalized log-likelihood score. *IEEE Trans on Speech and Audio Processing*, 4(1):56–60, 1996.

[8] W. D. Zhang, M. W. Mak, C. K. Li, and M. X. He. A priori threshold determination for phrase-prompted speaker verification. In *Eurospeech'99*, volume 2, pages 1023–1026, 1999.

[9] Jr. J. P. Campbell. Testing with the YOHO CD-ROM voice verification corpus. In *ICASSP'95*, pages 341–344, 1995.

[10] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET Curve in assessment of detection task performance. In *Eurospeech'97*, pages 1895–1898, 1997.