ON THE USE OF PCA IN GMM AND AR-VECTOR MODELS FOR TEXT INDEPENDENT SPEAKER VERIFICATION

Charles B. de Lima[1*], *Abraham Alcaim*[2], *and José A. Apolinário Jr.*[1]

[1] IME - Department of Electrical Engineering, Praça General Tibúrcio, 80—Urca, 22.290-270 Rio de Janeiro, RJ, Brazil
cborges/apolin@epq.ime.eb.br
[2]CETUC/PUC-Rio, Rua Marquês de São Vicente, 225—Gávea, 22453-900, Rio de Janeiro, RJ, Brazil
alcaim@cetuc.puc-rio.br

Abstract: This paper examines the role of the Principal Components Analysis (PCA) on the performance of two classification systems for text independent speaker verification: the Gaussian Mixture Model (GMM) and the AR-Vector Model. The use of the PCA transform resulted in an improvement in the performance of the GMM for training times of $60s$ and $30s$. However, the advantage of using PCA was not observed for the AR-Vector model. For the case of $10s$ training time, there was no benefit in using PCA even with GMM. In this situation, the AR-Vector is superior for a $10s$ test and worse for a $3s$ test. In this latter case, however, all systems yielded error rates above 7%.

1. INTRODUCTION

Speech, being present everywhere from telephone nets to personal computers, may be the cheapest form to supply a growing need of providing authenticity and privacy in the worldwide communication nets [1]. Speaker verification is the task of verifying if a speech signal (utterance) belongs or not to a certain person, which means a binary decision. The decisions are carried out in the so-called speakers open set [2] because the recognition is done in an unknown speakers set (possible impostors). As to text dependency, recognition can be dependent or independent. Systems demanding a predetermined word or phrase are text dependent.

Two classification systems using PCA are investigated in this paper: the GMM and the AR-Vector. The GMM [3] combines the robustness and smoothing properties of the parametric Gaussian model with the arbitrary modeling capability of a non-parametric VQ. The GMM can also be understood as a single state HMM (Hidden Markov Model), having as observations mixtures of Gaussian PDFs (probability density functions). These components may model a vast phonetic class to characterize the sound produced by a person [4].

The AR-Vector—AR from *Auto-Regressive*—is a model capable of capturing information about the dynamics of the speech for a given speaker which is interpreted as the speaker articulatory capacity or, in other words, the way he (or she) speaks as time goes by [5]. In speaker recognition applications, the AR-Vector uses a distance measure in order to compare two models.

The Principal Components Analysis (PCA) is used with the purpose of decorrelating the training data. This leads to easier statistical models and adds structural information from the training data (eigenvalues of the covariance matrix) in an effort to provide more discriminative features to the speaker recognition system.

This paper is organized as follows. Section 2 and 3 briefly review the GMM and the AR-Vector, respectively. The PCA is presented in Section 4. Section 5 contains details of the system setup and presents the simulation results. Concluding remarks are given in Section 6.

2. THE GAUSSIAN MIXTURE MODEL

A mixture of Gaussian probability densities is a weighted sum of M densities, and is given by $p(\vec{x}|\lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x})$ where $\vec{x}$ is a $D \times 1$ random vector, $b_i(\vec{x})_{i=1,\ldots,M}$ are the density components, and $p_{i\,i=1,\ldots,M}$ are the mixtures weights.

Each component density is a D variate Gaussian function with mean vector $\vec{\mu}_i$ and covariance matrix $\mathbf{K}_i$. The complete Gaussian mixture density is parameterized by mean vectors, covariance matrices, and a weighted mixture of all component densities (λ model). These parameters are jointly represented by the notation $\lambda = \{p_i, \vec{\mu}_i, K_i\}, i = 1, ..., M$.

*The author thanks CAPES, Brazil, for partial funding of this work.

0-7803-7503-3/02/$17.00 ©2002 IEEE

For a set of training data, the model parameters are determined in order to maximize the likelihood of the GMM.

The algorithm presented in [3] is widely used for this task. For a sequence of T independent training vectors $X = \{\vec{x}_1, ..., \vec{x}_T\}$, the likelihood of the GMM for modeling a true speaker (model λ) is calculated through $\log p(X|\lambda) = \frac{1}{T}\sum_{t=1}^{T} \log p(\vec{x}_t|\lambda)$. The scale factor $\frac{1}{T}$ is used in order to normalize the likelihood according to the duration of the utterance (number of feature vectors).

The speaker verification system requires a binary decision, accepting or rejecting a pretense speaker. Such a system is represented in Fig. 1.

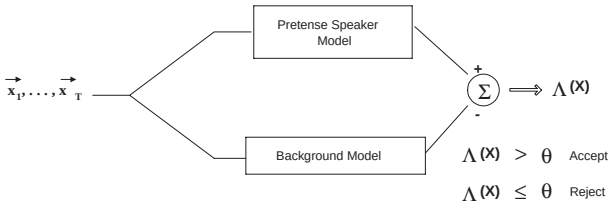

Fig. 1. Speaker verification system using GMM.

The system uses two models which provide the normalized logarithmic likelihood with input vectors $\vec{x}_1, ..., \vec{x}_T$, one from the pretense speaker and another one trying to minimize the variations not related to the speaker (**background** model), providing a more stable decision threshold [2]. If the system output value (difference between the two likelihoods) is higher than a given threshold θ the speaker is accepted; otherwise it is rejected. The background is built with a hypothetical set of false speakers and modeled via GMM (universal background model [6]). The threshold is calculated on the basis of experimental results.

3. THE AR-VECTOR MODEL

The AR-Vector is actually an extension of the LPC in the sense that it carries out a prediction among vectors (not samples), modeling the time evolving of the vectors (in our case, the feature vectors of speech). The order p AR-Vector model for a sequence of N vectors of dimension $m \times 1$, in time domain, is given by $\mathbf{X}_n = \sum_{k=1}^{p} \mathbf{A}_k X_{n-k} + \mathbf{E}_n$, where $\mathbf{X}_n$ and $\mathbf{E}_n$ are dimension $m \times 1$ vectors, with $\mathbf{E}$ representing the linear prediction error, and $\mathbf{A}_k$ being an $m \times m$ prediction matrix. The set of prediction matrices can be represented by an $m \times (p+1)$ matrix $\mathbf{A} = [\mathbf{A}_0 \quad \mathbf{A}_1 \quad \mathbf{A}_2 \quad \cdots \quad \mathbf{A}_p]$, with $\mathbf{A}_0 = \mathbf{I}$ (identity matrix).

From the vectors $\mathbf{X}_n$, we can define an estimate of the autocorrelation matrix $\mathbf{R}_k = \sum_{n=0}^{N-k} \mathbf{X}_n \mathbf{X}_{n+k}^T$, where N is the number of vectors $\mathbf{X}_n$ available for the estimation. Note that $\mathbf{R}_k$ results in a $m \times m$ matrix.

$\mathbf{A}_k$ are obtained by solving the following set of equations:

$$\begin{pmatrix} \mathbf{R}_0 & \mathbf{R}_1^T & \cdots & \mathbf{R}_{p-1}^T \\ \mathbf{R}_1 & \mathbf{R}_0 & \cdots & \mathbf{R}_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{p-1} & \mathbf{R}_{p-2} & \cdots & \mathbf{R}_0 \end{pmatrix} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_p \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_p \end{pmatrix} \tag{1}$$

From the previous equation, if we define the $pM \times pM$ Toeplitz autocorrelation matrix as $\boldsymbol{\mathcal{R}}$, the $pM \times M$ coefficient matrix as $\boldsymbol{\mathcal{A}}$, and the $pM \times M$ autocorrelation matrix on the right-hand side as $\mathbf{R}$, we have $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{R}}^{-1}\mathbf{R}$. Once $\boldsymbol{\mathcal{R}}$ is a Toeplitz matrix, a well known computationally efficient algorithm (Levinson-Durbin recursion) can be used to solve the set of equations.

The utilization of the AR-Vector in speaker recognition requires the use of some measure to evaluate the similarity between two autoregressive models. The use of the Itakura distance with the AR-Vector is presented in [5]. Assuming a stored model $\boldsymbol{\mathcal{A}}$ previously estimated from a given speaker and a model $\boldsymbol{\mathcal{B}}$ from a pretense speaker, three distance measures between these two models are defined for their respective autocorrelation matrices. In this work we will employ the symmetric distance, defined by

$$d_{\text{sym}} = \frac{1}{2}\left\{\log\left[\text{tr}\left(\frac{\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{R}}_B\boldsymbol{\mathcal{A}}^T}{\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{R}}_B\boldsymbol{\mathcal{B}}^T}\right)\right] + \log\left[\text{tr}\left(\frac{\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{R}}_A\boldsymbol{\mathcal{B}}^T}{\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{R}}_A\boldsymbol{\mathcal{A}}^T}\right)\right]\right\} \tag{2}$$

The speaker verification system provides a binary output, acceptance or rejection of a pretense speaker. Hence, an estimation of a threshold θ, based on true and false utterances, is required. This threshold is estimated with the *true distances*, i.e., the two models under comparison are from the same person, and with the *false distances* given by the pretense speaker model compared to the other models not belonging to him.

From these distances, the threshold is estimated taking into account false acceptance errors and false rejection errors. When a speaker is to be analyzed, he (or she) will be accepted if the resulting distance is lower than the threshold. He (or she) will be rejected otherwise. Fig. 2 presents the AR-Vector verification system.

The autoregressive model produces a smoothed model of the evolving features, capturing information from the dynamics of the speaker.

4. PRINCIPAL COMPONENTS ANALYSIS

Principal Components Analysis (PCA) is a linear mapping technique widely used in pattern recognition [7]. This technique extracts the features of a random vector from its projection over a set of base vectors. Considering a dimension $N \times 1$ random vector $\mathbf{X}$, the set of features $\mathbf{Y}$ can be extracted from $\mathbf{X}$ as follows:

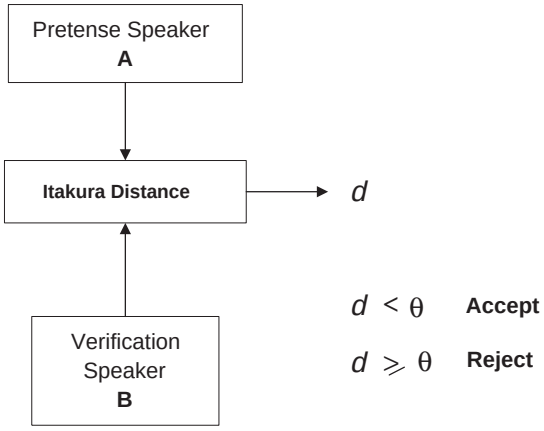

Fig. 2. AR-Vector Speaker Verification System.

$\mathbf{Y} = \mathbf{\Phi}^T\mathbf{X}$, where $\mathbf{\Phi}$ is a matrix formed by the base vectors, $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_N$, which extracts the features from $\mathbf{X}$ via the linear combination of its components. If these base vectors $\boldsymbol{\phi}_i$ are the eigenvectors of the covariance matrix of $\mathbf{X}$, then the resulting feature extraction is called *principal component analysis* [8]. This procedure also corresponds to the discrete version of the *Karhunen-Loéve Transform*.

The base vectors $\boldsymbol{\phi}_i$ can be used to represent $\mathbf{X}$ in a dimension M lower than the original ($M < N$). Once $\boldsymbol{\phi}_i$ are mutually orthogonal, $\mathbf{X}$ can be reconstructed from $\mathbf{Y}$ using the following expression: $\hat{\mathbf{X}} = \mathbf{\Phi}\mathbf{Y}$.

In PCA, the eigenvectors $\boldsymbol{\phi}_i$ are organized in the transformation matrix $\mathbf{\Phi}$ in such a way that all eigenvectors indices are in a descending order corresponding to their respective eigenvalues ($v_1 \geq v_2 \geq \cdots \geq v_N$). Once $\boldsymbol{\phi}_i$ are the eigenvectors of the covariance matrix of $\mathbf{X}$, then the reconstruction mean square error is minimal for any given M. The transformation performed by matrix $\mathbf{\Phi}$ will force the decorrelation of the random variables of the transformed vector $\mathbf{Y}$, resulting in a diagonal covariance matrix.

The use of PCA in this work aims at the generation of diagonal covariance matrices, with decorrelated data and without features dimension reduction. Each speaker will have an associated transformation matrix. Therefore, each of these transformation matrices—e.g. $\mathbf{\Phi}_L$ for speaker L—will be dependent on the covariance matrix of its correspondent speaker. The speaker transformation matrix $\mathbf{\Phi}_L$ is generated during the training phase and will be stored for the transformation to be carried out during the test phase.

5. SYSTEM SETUP AND SIMULATION RESULTS

This section details the setup and the results of the speaker verification system implemented in our experiments. The utterances were recorded with $8KHz$ as sampling rate, electret microphones, and in a low noise environment. We have used 36 speakers, 23 males and 13 females, from which 5 males (M) and 5 females (F) were selected exclusively to form the background and, therefore, did not participate in the tests. Each speaker uttered 200 sentences, in Brazilian Portuguese, extracted from [9]. We have used 15 mel-cepstrum coefficients (MCC) [10], with $20ms$ windows and 50% overlapping. The silence between words were eliminated. The number of Gaussians for the GMM was set to 32 while AR-Vector used order 2 with the symmetric Itakura distance (previous experiments have shown its better performance for this configuration). We have used 60, 30, and $10s$ of speech signal for training and 30, 10, and $3s$ for testing. Each background speaker contributed with 6 seconds of speech (without silence). The setting of the decision threshold was established in order to equally minimize the error rate between false acceptance—FA (to accept someone which does not correspond to the true speaker)—and false rejection—FR (to reject someone which corresponds to the true speaker). This procedure resulted in an equal error rate (EER) measure [2].

The results obtained with the 32 Gaussians GMM will be compared to the order 2 AR-Vector using symmetric Itakura distance, using both MCC and MCCPCA (PCA transformed MCC vectors). The performance with $60s$ of training can be seen in Table 1. For $30s$ test, both classification systems have presented no errors. We can observe that the use of PCA on the MCC feature vectors resulted in an improvement which was more visible for the GMM than for the AR-Vector. Moreover, the performance of the GMM was better than the performance of the AR-Vector for $10s$ and $3s$ of testing time, mainly in the latter case. With only $3s$ test there is not enough amount of data for an adequate modeling of the AR-Vector which produced errors at a rate around 10%.

Table 1. Performance of the GMM versus the AR-Vector, with and without PCA, for $60s$ training.

System	tests(%)		
	30s	10s	3s
	EER	EER	EER
GMM - MCC	0	0.44	1.38
GMM - MCCPCA	0	0.38	1.23
AR-Vector - MCC	0	1.22	10.00
AR-Vector - MCCPCA	0	1.21	9.98

Table 2 presents the results for $30s$ of training. These results show that the PCA technique still favors the GMM but the same is not true for the AR-Vector. For $30s$ test, the AR-Vector has presented no errors, 1.15% better than the GMM with PCA. With $10s$ test, the performance of the AR-Vector is approximately the same as the GMM with MCC but inferior to the GMM with MCCPCA. For $3s$ test, the performance of the AR-Vector is almost 3 times lower than the GMM.

In Table 3 the results for the lowest training time are presented. They resulted in the highest error rates of both classification systems. When the training time is $10s$, the use of the PCA presented no significant improvement—and eventually loss of performance. The AR-Vector presented better results as compared to the GMM for the case of $10s$ test, but with an error rate around 4% higher for $3s$ test.

Table 2. Performance of the GMM versus the AR-Vector, with and without PCA, for $30s$ training.

System	tests(%)		
	30s	10s	3s
	EER	EER	EER
GMM - MCC	1.23	1.54	3.08
GMM - MCCPCA	1.15	1.28	2.73
AR-Vector - MCC	0	1.60	10.25
AR-Vector - MCCPCA	0	1.60	10.34

Table 3. Performance of the GMM versus the AR-Vector, with and without PCA, for $10s$ training.

System	tests(%)	
	10s	3s
	EER	EER
GMM - MCC	4.57	7.25
GMM - MCCPCA	4.47	7.39
AR-Vector - MCC	3.20	11.85
AR-Vector - MCCPCA	3.24	11.84

Throughout the analysis of the results presented here, we can clearly note that the amount of time for training and for testing has a strong influence. The larger they are the more statistics they are offering and, consequently, the more precise the modeling carried out by the GMM and AR-Vector will be. The use of PCA resulted in an improvement in the performance for the GMM, specially for higher training times. In the case of the AR-Vector, however, the improvement in performance, when existing, was negligible.

6. CONCLUDING REMARKS

This paper presented the performance of GMM and AR-Vector using PCA in text independent speaker verification systems. The results have shown the efficiency of both classification systems for different training and testing times. The best performance (no errors) with the lower computational complexity was obtained with the MCC AR-Vector procedure with $30s$ for training and testing.

The use of PCA and GMM provided performance gains for the highest training times. The best performance with the lowest testing times (10 and $3s$) was obtained with the MCCPCA-GMM, with 60 and $30s$ of training, and with errors from 0.38 to 2.73%.

The best performance with the lowest time of training and test corresponded to the AR-Vector with $10s$ training and testing—with errors around 3.2%. The use of PCA introduced no improvement with $10s$ training time for any of the two classification models.

7. REFERENCES

[1] CAMPBELL, Joseph P.,Jr. *Speaker Recognition: A Tutorial.* Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, September 1997.

[2] REYNOLDS, Douglas A. *Speaker Identification and Verification Using Gaussian Mixture Speaker Models.* Speech Communication, vol. 17, pp. 91-108, 1995.

[3] REYNOLDS, Douglas A. *A Gaussian Mixture Modeling Approach to Text Independent Speaker Identification.* PhD Thesis. Georgia Institute of Technology, August 1992.

[4] REYNOLDS, Douglas A. *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Model.* IEEE Transactions on Speech and Audio Processing, vol. 3, no. 1, pp. 72-83, January 1995.

[5] BIMBOT, F., L. Mathan, A. de Lima, and G. Chollet. *Standard and Target Driven AR-vector Models for Speech Analysis and Speaker Recognition*. Proceedings of ICASSP, San Francisco, USA, vol. 2, pp. II5-II8, March 1992.

[6] REYNOLDS, Douglas A. Thomas F. Quatieri, and Robert B. Dunn. *Speaker Verification Using Adapted Gaussian Mixture Models.* Digital Signal Processing, vol. 10, pp. 19-41, 2000.

[7] MALAYATH, Narendranath. *Data-driven Methods for Extracting Features From Speech.* PhD Thesis - Oregon Graduate Institute, 2000.

[8] FUKUNAGA, Keinosuke. *Introduction to Statistical Pattern Recognition*. 2nd. ed. USA: Morgan Kaufmann, 1990.

[9] ALCAIM, Abraham, José Alberto Solewicz, and João Antonio de Morais. *Freqüência de ocorrência dos fonemas e listas de frases foneticamente balanceadas no Português falado no Rio de Janeiro.* Revista da Sociedade Brasileira de Telecomunicações, vol. 7, no. 1, December 1992.

[10] DAVIS, Steven B., and Paul Mermelstein. *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences.* IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-28, no. 4, August 1980.