GMM VERSUS AR-VECTOR MODELS FOR TEXT INDEPENDENT SPEAKER VERIFICATION

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

[†]IME - Department of Electrical Engineering
Praça General Tibúrcio, 80 – Urca
22.290-270 Rio de Janeiro, RJ, Brazil
cborges/apolin@epq.ime.eb.br

[‡]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 presents a performance evaluation of two classification systems for text independent speaker verification: the Gaussian Mixture Model (GMM) and the AR-Vector Model. For the GMM, 32, 16, and 8 Gaussians are evaluated. On the other hand, an order 2 model with the Itakura symmetric distance was used for the AR-Vector. Both classification systems presented no errors when training and testing times were not smaller than 60s and 30s, respectively. Using 10s as the test time, the most accurate classification systems errors were between 0.4 and 3.3%. With 3s test, the errors presented by the GMM were around 6 to 7% whereas those for the AR-Vector were above 10%. However, the best results using 10s as testing and training times were obtained with the AR-Vector, with errors around 3.2%.

1. INTRODUCTION

The recognition of a human being through his voice is one of the simplest forms of automatic recognition because it uses biometric characteristics which come from a natural action, the speech. 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].

Research in the area of speaker recognition has significantly grown over the last few years due to a vast area of applications where the recognition can be used such as

– Access control: to devices, networks, and data in general;

– Authentication for business transactions as a tool to prevent fraud in: shopping over telephone, credit card validation, transactions over Internet, bank operations, etc.

– Law enforcement: penitentiary monitoring, forensic applications, etc.

– Help to handicapped.

– Military use: classified information requiring speaker identification.

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. Such systems can offer precise and reliable comparisons between two speech signals with the same text, in phonetically similar environments, requiring only 2 to 3 seconds of speech for training and testing. In text independent systems, such comparisons are not so easy to be obtained. The performance decreases as compared to text dependency. Moreover, in order to obtain reasonable statistics of the signal, it is, in general, necessary from 10 to 30 seconds of speech signal for training and testing [3].

In speaker recognition, the Gaussian Mixture Model (GMM) can be seen as a hybrid between two effective models: a unimodal Gaussian classifier and a vector quantization (VQ) codebook [4]. This scheme combines the robustness and smoothing properties of the parametric Gaussian model with the arbitrary modeling capability of a non-parametric VQ. The GMM performs the spatial separation of acoustic classes and its main difference comparing to VQ concerns the fact that distances are not used to separate classes but probabilities from a set of Gaussian probability density functions previously estimated. The GMM can also be understood as a single state HMM (Hidden Markov Model) [5], having as observations mixtures of Gaussian PDFs (probability density functions). These components may model a vast phonetic class to characterize the sound

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

produced by a person [6]. This fact justifies its use in speaker recognition.

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 speaks as time goes by [7]. This is an extension of a model widely known in speech processing, the Linear Prediction Coefficients (LPC). Whilst LPC is based on the linear regression over scalars, AR-Vector is based on the regression over feature vectors. In speaker recognition applications, the AR-Vector uses a distance measure in order to compare two models. For this measure, the so-called Itakura distance [8] is usually employed .

This paper is organized as follows. In Section 2, the GMM is reviewed. The AR-Vector is described in Section 3.. Section 4 contains details of the system consifguration and is followed by simulation results in Section 5, and conclusions 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

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \qquad (1)$$

where $\vec{x}$ is a random vector of dimension D, $b_i(\vec{x})$, $i = 1, ..., M$, are the density components, and p_i, $i = 1, ..., M$, are the mixtures weights. Each component density is a D variate Gaussian function of the form

$$b_i(\vec{x}) = \frac{e^{\left(-\frac{1}{2}(\vec{x}-\vec{\mu})' K_i^{-1}(\vec{x}-\vec{\mu})\right)}}{(2\pi)^{\frac{D}{2}}\sqrt{|\mathbf{K}_i|}} \qquad (2)$$

with mean vector $\vec{\mu}_i$ and covariance matrix $\mathbf{K}_i$.

Note that the weighting of the mixtures satisfies $\Sigma_{i=1}^{M} p_i = 1$. The complete Gaussian mixture density is parameterized by a vector of means, covariance matrix, and a weighted mixture of all component densities (λ model). These parameters are jointly represented by the following notation:

$$\lambda = \{p_i, \vec{\mu}_i, K_i\} \qquad i = 1, ..., M. \qquad (3)$$

The GMM can have different forms depending on the choice of the covariance matrix. The model can have a covariance matrix per Gaussian component as indicated in (3) (nodal covariance), a covariance matrix for all Gaussian components for a given model (grand covariance), or only one covariance matrix shared by all models (global covariance). A covariance matrix can also be complete or diagonal [2].

For a set of training data, the estimation of the maximum likelihood is necessary. In other words, this estimation tries to find the model parameters that maximize the likelihood of the GMM, The algorithm presented in [4] 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 is given by

$$p(X|\lambda) = \prod_{t=1}^{T} p(\vec{x}_t|\lambda) \qquad (4)$$

The likelihood for modeling a true speaker (model λ) is directly calculated through

$$\log p(X|\lambda) = \frac{1}{T}\sum_{t=1}^{T} \log p(\vec{x}_t|\lambda) \qquad (5)$$

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 last equation corresponds to the normalized logarithmic likelihood which is the λ model's response.

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

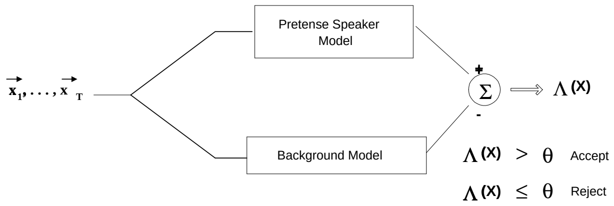
$\vec{x}_1, \ldots, \vec{x}_T$
Pretense Speaker Model
Background Model
+
-
Σ
$\Longrightarrow \Lambda(X)$
$\Lambda(X) > \theta$ Accept
$\Lambda(X) \leq \theta$ Reject

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 likelihood) 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 [9]). The threshold is calculated on the basis of experimental results.

3. AR-VECTOR

The basic idea behind linear prediction is that a speech sample can be approximated by a linear combination of p past samples. LPC is calculated from the samples of a numerical sequence (pieces of speech). 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.

The order p AR-Vector model for a sequence of N vectors of dimension m, in time domain, is given by:

$$X_n = \sum_{k=1}^{p} A_k X_{n-k} + E_n \tag{6}$$

where X_n and E_n are m dimension vectors, with E representing the linear prediction error, and A_k being the $m \times m$ prediction matrix. The set of prediction matrices can be represented by a $m \times (p+1)$ matrix $\mathbf{A} = [A_0 \quad A_1 \quad A_2 \quad \cdots \quad A_p]$, with $A_0 = I$ (identity matrix).

From vectors X_n, we can define an estimate of the autocorrelation matrix:

$$R_k = \sum_{n=0}^{N-k} X_n X_{n+k}^T \tag{7}$$

where N is the number of vectors X. Note that R_k results in a $m \times m$ matrix.

A_k are obtained by solving the following set of equations.

$$\begin{pmatrix} R_0 & R_1^T & \cdots & R_{p-1}^T \\ R_1 & R_0 & \cdots & R_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{p-1} & R_{p-2} & \cdots & R_0 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_p \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_p \end{pmatrix} \tag{8}$$

From the previous equation, if we define the Toeplitz autocorrelation matrix as $\mathbf{R}$, the coefficient matrix as $\mathbf{A}$, and the autocorrelation matrix on the right-hand side as R, we have:

$$\mathbf{RA} = \mathrm{R} \quad \Rightarrow \quad \mathbf{A} = \mathbf{R}^{-1}\mathrm{R} \tag{9}$$

Once $\mathbf{R}$ is a Toeplitz matrix, a well known computationally efficient algorithm (the Levinson-Durbin recursion) can be used to solve the set of equations [10].

The utilization of the AR-Vector in speaker recognition requires the use of some measure to evaluate the similarity between two autoregressive models. A widely used distance measure is the Itakura distance [8] which provides the distance between two all-poles LPC's based on the linear prediction coefficients and on the autocorrelation matrix.

The use of the Itakura distance with the AR-Vector is presented in [7]. Assuming a stored model $\mathbf{A}$ previously estimated from a given speaker and a model $\mathbf{B}$ from a pretense speaker, three distance measures between these two model are defined for their respective autocorrelation matrices. These measures are:

1. Distance from $\mathbf{B}$ to $\mathbf{A}$:

$$d(\mathbf{B}, \mathbf{A}) = \log(\mathrm{tr}\left[\frac{\mathbf{A}\mathbf{R_B}\mathbf{A}^T}{\mathbf{B}\mathbf{R_B}\mathbf{B}^T}\right]) \tag{10}$$

2. Distance from $\mathbf{A}$ to $\mathbf{B}$:

$$d(\mathbf{A}, \mathbf{B}) = \log(\mathrm{tr}\left[\frac{\mathbf{B}\mathbf{R_A}\mathbf{B}^T}{\mathbf{A}\mathbf{R_A}\mathbf{A}^T}\right]) \tag{11}$$

3. Symmetric Distance:

$$d_{\mathrm{sim}} = \frac{1}{2}(d(\mathbf{B}, \mathbf{A}) + d(\mathbf{A}, \mathbf{B})) \tag{12}$$

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 will be accepted if the resulting distance is lower than the threshold. He will be rejected otherwise. Fig. 2 presents the AR-Vector verification system.

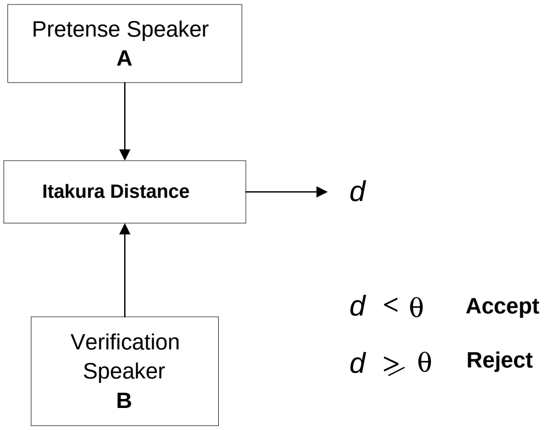

Fig. 2. AR-Vector Speaker Verification System.

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

4. SYSTEM CONFIGURATION

This section details the speaker verification system implemented in our experiments. 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 [11]. We have used 15 Mel-*cepstrum* coefficients [12], with $20ms$ windows and 50% overlapping. The silence between words were eliminated. The number of Gaussians were 32, 16, and 8. In the order 2 AR-Vector, we have used the symmetric Itakura distance because 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].

5. SIMULATION RESULTS

The results obtained with the GMM, varying the number of Gaussians, are compared with the results obtained with the order 2 AR-Vector, using symmetrical Itakura distance. The reason for this choice is due to the fact that we have run experiments with orders 2 to 5 and obtained similar results; order 2 was chosen for its lower computational complexity.

Table 1. Performance Results, GMM × AR-Vector, for $60s$ of training.

System	tests results (%)		
	30s	10s	3s
	EER	EER	EER
GMM - 32 G	0	0.44	1.38
GMM - 16 G	0.50	0.53	2.03
GMM - 8 G	1.34	1.92	3.59
AR-Vector	0	1.22	10.00

The results with $60s$ of training are shown in Table 1. From this table we see that, for $30s$ test, the AR-Vector's results were superior than the GMM's results with 16 and 8 Gaussians. For $10s$ test, nevertheless, the AR-Vector resulted superior with respect only to the 8 Gaussians GMM. In this case, the AR-Vector yields an EER more than double of the one obtained with the GMM using 32 and 16 Gaussians. Finally, with $3s$ test, the AR-Vector presented a result much worser than the GMM.

Table 2 presents the results for a $30s$ training time.

In Table 2 the AR-Vector overcame the GMM for $30s$ test, presenting no errors. With $10s$ test, the AR-Vector presented superior results than the 16 and 8 Gaussians GMM, and is close to the results of the GMM with 32 Gaussians. With $3s$ test, the AR-Vector presented errors around 10% against the 3% to 7% of the GMM.

Table 2. Performance Results, GMM × AR-Vector, for $30s$ of training.

System	tests results (%)		
	30s	10s	3s
	EER	EER	EER
GMM - 32 G	1.53	1.54	3.08
GMM - 16 G	3.08	3.30	4.85
GMM - 8 G	4.60	5.30	6.80
AR-Vector	0	1.60	10.25

In Table 3 we find the results for the lower training time, corresponding to the worst errors of the classification systems.

Table 3. Performance Results, GMM × AR-Vector, for $10s$ of training.

System	tests results (%)	
	10s	3s
	EER	EER
GMM - 32 G	4.57	7.25
GMM - 16 G	4.87	7.00
GMM - 8 G	5.25	7.17
AR-Vector	3.2	11.85

When the training time is $10s$, the GMM—independently of the number of Gaussians—yields results which are very close to each other. The AR-Vector achieved better results only for the $10s$ test case (errors 1 to 2% lower).

Throughout the analysis of the results presented here, we can clearly note that the number of Gaussians has a strong influence in the performance. The higher this number the better the modeling obtained by the GMM and, therefore, the better the results will be. The amount of time for training and for testing also have 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. When the statistics provided to train the GMM is poor, the number of Gaussians does not influence the response because there is no data for a more precise modeling, as can be observed in Table 3. The same is valid for the AR-Vector which is not able to offer an adequate modeling when only $3s$ of data is available for testing.

6. CONCLUSIONS

This paper compared the performances of the GMM versus the AR-Vector models for text independent speaker verification. The results have shown the efficiency of both schemes for different times of training and testing as well as for different number of Gaussians in the GMM. Based on the results available in this paper, we can conclude that:

- The best performance (no errors) with the lower computational complexity was obtained by the AR-Vector procedure with $30s$ for training and testing.

- The best performance for the lowest testing times (10 and $3s$) was obtained with the 32 Gaussians GMM, with 60 and $30s$ of training, and with errors from 0.5 to 3%. Yet for $60s$ training, the GMM with 16 Gaussians presented errors from 0.5 to 2%, overcoming the AR-Vector.

- 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%.

7. REFERENCES

[1] CAMPBELL, Joseph P., Jr. *Speaker Recognition: A Tutorial.* Proceedings of 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] JAYANT M. Naik. *Speaker Verification: A Tutorial.* IEEE Communication Magazine, pp. 42-47, January 1990.

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

[5] RABINER, Lawrence R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of The IEEE, vol. 77, no. 2, February 1989.

[6] REYNOLDS, Douglas A. *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Model.* IEEE Transactions on Speech and Audio Processing. vol. 3, n. 1, pp. 72-83, January, 1995.

[7] 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, v. 2, p. II5-II8, March 1992.

[8] ITAKURA, Fumitada. *Minimum Prediction Residual Principle Applied to Speech Recognition*. IEEE Transactions on Acoustics, Speech, and Signal Processing, v. ASSP-23, n. 1, February 1975.

[9] 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.

[10] HAYKIN, Simon. Adaptive Filter Theory. 3. ed. New Jersey: Prentice Hall, 1996.

[11] 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, nr 1, December 1992.

[12] 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.