State of the Art: Speech Biometrics Verification

Mohamed Soltane, Noureddine Doghmane, Nourddine Guersi
Electronics Department, Faculty of engineering science
Badji Mokhtar University of Annaba
23000 Annaba
Algeria
soltane@lri-annaba.net, xor99@hotmail.com, ndoghmane@univ-annaba.org, guersi54@yahoo.fr

ABSTRACT: *Gaussian mixture models (GMMs) remain the state of the art technique for modeling spectral envelope features for speech recognition systems. This paper presents a comparative analysis of the performance of three estimation algorithms Expectation Maximization (EM), Greedy EM Algorithm (GEM) and Figueiredo-Jain Algorithm (FJ) based Gaussian mixture models (GMMs) for text-independent speech biometrics verification. The simulation results are showed significant performance achievements. The test performance of, EER=0.26 % for "EM", EER=0.21 % for "GEM" and EER=0.16 % for "FJ", show that the behavioral information scheme of speech biometrics is more robust and have a discriminating power, which can be explored for identity authentication.*

Keywords: Biometric authentication, Behavioral, Speech, Soft decision and Gaussian Mixture Modal, EM, GEM and FJ

Received: 12 September 2009, Revised 13 November 2009, Accepted 11 December 2009

1. Introduction

BIOMETRIC is a Greek composite word stemming from the synthesis of bio and metric, meaning life measurement. In this context, the science of biometrics is concerned with the accurate measurement of unique biological characteristics of an individual in order to securely identify them to a computer or other electronic system. Biological characteristics measured usually include fingerprints, voice patterns, retinal and iris scans, face patterns, and even the chemical composition of an individual's DNA [1]. Biometrics authentication (BA) (*Am I whom I claim I am?*) involves confirming or denying a person's *claimed identity* based on his/her physiological or behavioral characteristics [2]. BA is becoming an important alternative to traditional authentication methods such as keys ("something one has", i.e., by possession) or PIN numbers ("something one knows", i.e., by knowledge) because it is essentially "who one is", i.e., by biometric information. Therefore, it is not susceptible to misplacement or forgetfulness [3]. These biometric systems for personal authentication and identification are based upon physiological or behavioral features which are typically distinctive, although time varying, such as fingerprints, hand geometry, face, voice, lip movement, gait, and iris patterns. An identity verification system has to deal with two kinds of events: either the person claiming a given identity is the one who he claims to be (in which case, he is called a *client*), or he is not (in which case, he is called an *impostor*). Moreover, the system may generally take two decisions: either *accept* the *client* or *reject* him and decide he is an *impostor*.

Some works based on biometric speech identity verification systems has been reported in literature. **Fortuna J. et al. [16]** present a comparative analysis of the performance of decoupled and adapted Gaussian mixture models (GMMs) for open-set, text-independent speaker identification (OSTISI) and concluded that the speaker identification performance is noticeably better with adapted-GMMs than with decoupled-GMMs. Their analysis is based on a set of experiments using an appropriate subset of the NIST-SRE 2003 database and various score normalization methods. They included a detailed description of the experiments and discuss how the OSTI-SI performance is influenced by the characteristics of each of the two modeling techniques and the normalization approaches adopted. **Eduardo Sànchew-Soto et al. [17]** present a new adaptation technique for speaker verification of models built using Bayesian Networks tested using the NIST 2002 data base and showed improvement in the verification performances. The adaptation problem of parameters of the conditional probability tables (CPTs) is treated in a specific manner. The model adaptation involves estimating the new vectors with a transformation that includes vectors in the world model and the speaker model and the combination of both models is based on a value computed using a measure of distance between vectors of both CPTs. **Arnon Cohen et al. [18]** describe an HMM based speaker verification system evaluated on a text-dependent database, which verifies speakers in their own specific feature space. The user feature space is determined by a Dynamic Programming (DP) feature selection algorithm, in which a suitable criterion, correlated with Equal Error Rate (EER) was developed and is used for this feature selection algorithm. A significant improvement in verification results was demonstrated with the DP selected individual feature space. An EER of 4.8% was achieved when the feature set was the “almost standard” Mel Frequency Cepstrum Coefficients (MFCC) space (12 MFCC + 12 ΔMFCC). Under the same conditions, a system based on the selected feature space yielded an EER of only 2.7%. **Mijail Arcienega et al. [19]** present a Bayesian network approach for modeling the pitch and spectral envelope and showed an increase in the performance of the speaker recognition system. In which the conditional statistical distributions (represented by GMMs) of the features are simultaneously exploited for increasing the recognition score within the approach, and in particularly in noisy conditions. **Driss Matrouf et al. [20]** investigate the effect of voice transformation on automatic speaker recognition systems performance and showed an increase of about 2.7 time of the likelihood ratio, without a degradation of the natural aspect of the voice. It focuses on increasing the impostor acceptation rate, by modifying the voice of an impostor in order to target a specific speaker. Their work is inspired from the idea that in several forensic situations, it is reasonable to think that some organizations have a knowledge on the speaker recognition method used by the police department and could impersonate a given, well known speaker.

2. Biometric Speech Verification

2.1 Speech Analysis and Feature Extraction

Gaussian Mixture Models (GMMs), is the main tool used in text-independent speaker verification, in which can be trained using the Expectation Maximization (EM) algorithm [4]. In this work the speech modality, is authenticated with a multi-lingual text-independent speaker verification system. The speech trait is comprised of two main components as shown in figure 1: speech feature extraction and a Gaussian Mixture Model (GMM) classifier. The speech signal is analyzed on a frame by frame basis, with a typical frame length of 20 ms and a frame advance of 10 ms [5]. For each frame, a dimensional feature vector is extracted, the discrete Fourier spectrum is obtained via a fast Fourier transform from which magnitude squared spectrum is computed and put it through a bank of filters. The critical band warping is done following an approximation to the Mel-frequency scale which is linear up to 1000 Hz and logarithmic above 1000 Hz. The Mel-scale cepstral coefficients are computed from the outputs of the filter bank [6]. The state of the art speech feature extraction schemes (Mel frequecy cepstral coefficients (MFCC) is based on auditory processing on the spectrum of speech signal and cepstral representation of the resulting features [7]. One of the powerful properties of cepstrum is the fact that any periodicities, or repeated patterns, in a spectrum will be mapped to one or two specific components in the cepstrum. If a spectrum contains several harmonic series, they will be separated in a way similar to the way the spectrum separates repetitive time patterns in the waveform. The description of the different steps to exhibit features characteristics of an audio sample with MFCC is showed in figure 2.

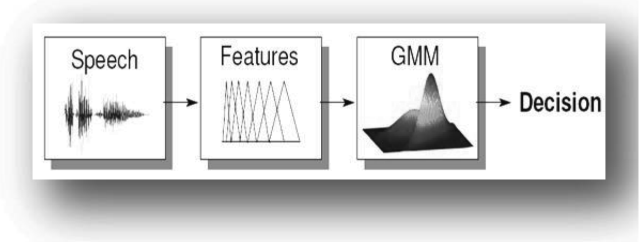


Figure 1. Acoustic Speech Analysis

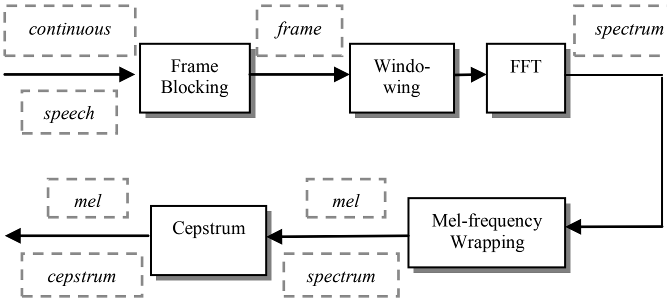


Figure 2. MFCC calculation Block diagram [6]

The distribution of feature vectors for each person is modeled by a GMM. The parameters of the Gaussian mixture probability density function are estimated using three different estimation algorithms. The Expectation Maximization (EM) algorithm [8], Greedy algorithm (GEM) [8] and Figueiredo-Jain (FJ) algorithm [8].

Given a claim for person ***C's*** identity and a set of feature vectors $X = \left\{\vec{x}_i\right\}_{i=1}^{N_V}$ supporting the claim, the average log likelihood of the claimant being the true claimant is calculated using:

$$\mathcal{L}(X|\lambda_C) = \frac{1}{N_V}\sum_{i=1}^{N_V}\log\ p(\vec{x}_i|\lambda_C) \tag{1}$$

where $$p(\vec{x}|\lambda) = \sum_{j=1}^{N_M} m_j\ \mathcal{N}\left(\vec{x}\ ;\vec{\mu_j};\Sigma_j\right) \tag{2}$$

and $$\lambda = \left\{m_j\ ,\ \vec{\mu_j}\ ,\Sigma_j\right\}_{j=1}^{N_M} \tag{3}$$

Here λ_C is the model for person *C*. N_M is the number of mixtures, m_j is the weight for mixture *j* (with constraint $\sum_{j=1}^{N_M} m_j = 1$), and $N(\vec{x}\ ;\vec{\mu}\ ,\Sigma)$ is a multi-variate Gaussian function with mean $\vec{\mu}$ and diagonal covariance matrix Σ. Given a set $\{\lambda_b\}_{b=1}^{B}$ of ***B*** background person models for person ***C***, the average log likelihood of the claimant being an impostor is found using:

$$\mathcal{L}\left(X|\lambda_{\bar{C}}\right) = \ \log\left[\frac{1}{B}\ \sum_{b=1}^{B}\exp\mathcal{L}(X|\lambda_b)\right] \tag{4}$$

The set of background person models is found using the method described in [15]. An opinion on the claim is found using:

$$o = \ \mathcal{L}(X|\lambda_C) - \mathcal{L}\left(X|\lambda_{\bar{C}}\right) \tag{5}$$

The opinion reflects the likelihood that a given claimant is the true claimant (i.e., a low opinion suggests that the claimant is an impostor, while a high opinion suggests that the claimant is the true claimant).

2.2 Maximum Likelihood Parameter Estimation

Given a set of observation data in a matrix X and a set of observation parameters θ the ML parameter estimation aims at maximizing the likelihood $L(\theta)$ or log likelihood of the observation data $X=\{X_1,\ldots,X_n\}$

$$\hat{\theta} = \ arg\ \max_{\theta} L(\theta). \tag{6}$$

Assuming that it has independent, identically distributed data, it can write the above equations as:

$$L(\theta) = p(X|\theta) = p(X_1,\ldots,X_n|\theta) = \prod_{i=1}^{n} p(X_i|\theta). \tag{7}$$

The maximum for this function can be find by taking the derivative and set it equal to zero, assuming an analytical function.

$$\frac{\partial}{\partial\theta}L(\theta) = 0. \tag{8}$$

The incomplete-data log-likelihood of the data for the mixture model is given by:

$$L(\theta) = log(X|\theta) = \sum_{i=1}^{N} log(x_i|\theta) \tag{9}$$

Which is difficult to optimize because it contains the log of the sum. If it considers X as incomplete, however, and posits the existence of unobserved data items $Y = \{y_i\}_{i=1}^{N}$ whose values inform us which component density generated each data item, the likelihood expression is significantly simplified. That is, it assume that $y_i \in \{1..K\}$ for each i, and $y_i = k$ if the i-th sample was generated by the k-th mixture component. If it knows the values of Y, it obtains the complete-data log-likelihood, given by:

$$L(\theta, Y) = \log p(X, Y|\theta) \tag{10}$$

$$= \sum_{i=1}^{N} \log p(x_i, y_i|\theta) \tag{11}$$

$$= \sum_{i=1}^{N} \log\big(p(y_i|\theta)p(x_i|y_i, \theta)\big) \tag{12}$$

$$= \sum_{i=1}^{N} \big(\log p_{y_i} + \log g\big(x_i|\mu_{y_i}, \textstyle\sum_{y_i}\big)\big) \tag{13}$$

Which, given a particular form of the component densities, can be optimized using a variety of techniques [9].

2.2.1 EM algorithm

The expectation-maximization (EM) algorithm [8, 10, 11, 12] is a procedure for maximum-likelihood (ML) estimation in the cases where a closed form expression for the optimal parameters is hard to obtain. This iterative algorithm guarantees the monotonic increase in the likelihood *L* when the algorithm is run on the same training database.

The probability density of the Gaussian mixture of *k* components in R^d can be described as follows:

$$\Phi(x) = \sum_{i=1}^{N} \pi_i \, \emptyset(x|\theta_i) \quad \forall x \in R^d, \tag{14}$$

where $\phi(x|\theta_i)$ is a Gaussian probability density with the parameters $\theta_i = (m_i, \sum_i)$is the mean vector and $\sum_i$ is the covariance matrix which is assumed positive definite given by:

$$\emptyset(x|\theta_i) = \emptyset(x|m_i, \textstyle\sum_i) = \frac{1}{(2\pi)^{\frac{n}{2}}|\sum_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-m_i)^T \sum_i^{-1}(x-m_i)}, \tag{15}$$

and $\pi_i \in [0, 1]$ ($i = 1,2,..., k$) are the mixing proportions under the constraint $\sum_{i=1}^{k} \pi_i = 1$. If it encapsulate all the parameters into one vector: $\Theta_k = (\pi_1, \pi_2,..., \pi_k, \theta_1, \theta_2, ..., \theta_k)$, then , according to Eq. (23), the density of Gaussian mixture can be rewritten as:

$$\Phi(x|\Theta_k) = \sum_{i=1}^{k} \pi_i \emptyset(x|\theta_i) = \sum_{i=1}^{k} \pi_i \emptyset(x|m_i, \textstyle\sum_i). \tag{16}$$

For the Gaussian mixture modeling, there are many learning algorithms. But the EM algorithm may be the most well-known one. By alternatively implementing the E-step to estimate the probability distribution of the unobservable random variable and the M-step to increase the log-likelihood function, the EM algorithm can finally lead to a local maximum of the log-likelihood function of the model. For the Gaussian mixture model, given a sample data set $S = \{x_1, x_2, \cdots, x_N\}$ as a special incomplete data set, the log-likelihood function can be expressed as follows:

$$\log p(S|\Theta_k) = \log \prod_{t=1}^{N} \emptyset(x_t|\Theta_k) = \sum_{t=1}^{N} \log \sum_{i=1}^{k} \pi_i \emptyset(x_t|\theta_i), \tag{17}$$

Which can be optimized iteratively via the EM algorithm as follows:

$$P(j|x_t) = \frac{\pi_j \emptyset(x_t|\theta_j)}{\sum_{i=1}^{k} \pi_i \emptyset(x_t|\theta_i)}, \tag{18}$$

$$\pi_j^+ = \frac{1}{N} \sum_{t=1}^{N} P(j|x_t), \tag{19}$$

$$\mu_j^+ = \frac{1}{\sum_{t=1}^{N} P(j|x_t)} \sum_{t=1}^{N} P(j|x_t) x_t, \tag{20}$$

$$\textstyle\sum_j^+ = \frac{1}{\sum_{t=1}^{N} P(j|x_t)} \sum_{t=1}^{N} P(j|x_t)(x_t - \mu_j^+)(x_t - \mu_j^+)^T. \tag{21}$$

Although the EM algorithm can have some good convergence properties in certain situations, it certainly has no ability to determine the proper number of the components for a sample data set because it is based on the maximization of the likelihood.

2.2.2 Greedy EM Algorithm

The greedy algorithm (GEM) [8, 10, 12, 13] starts with a single component and then adds components into the mixture one by one. The optimal starting component for a Gaussian mixture is trivially computed, optimal meaning the highest training data likelihood. The algorithm repeats two steps: insert a component into the mixture, and run EM until convergence.

Inserting a component that increases the likelihood the most is thought to be an eas ier problem than initializing a whole near-optimal distribution. Component insertion involves searching for the parameters for only one component at a time. Recall that EM finds a local optimum for the distribution parameters, not necessarily the global optimum which makes it initialization dependent method.

Given p_c a C-component Gaussian mixture with parameters θ_c. the general greedy algorithm for Gaussian mixture is as follows:

1. *Compute the optimal (in the ML sense) one-component mixture* p_1 *and set* C←1.
2. *Find a new component* $N(x\,;\,\mu',\,\Sigma')$ *and corresponding mixing weight* α' *that increase the likelihood the most:*

$$\{\mu',\Sigma',\alpha'\} = \arg\max_{\{\mu,\Sigma,\alpha\}} \sum_{i=1}^{N} \ln\left[(1-\alpha)p_C\,(x_i) + \alpha\mathcal{N}\left(x_{i;}\,\mu,\Sigma\right)\right] \tag{22}$$

while keeping p_c *fixed.*

3. *Set* $p_{c+1}\,(x) \leftarrow (1-\alpha')\,p_c\,(x) + \alpha'\,N(x\,;\,\mu',\,\Sigma')$

   and then $C \leftarrow C + 1$.
4. *Update* p_c *using* ***EM*** *(or more other method) until convergence.*
5. *Evaluate some stopping criterion; go to step 2 or quit.*

The stopping criterion in Step 5 can be for example any kind of model selection criterion, wanted number of components, or the minimum message length criterion.

The crucial point is of course Step 2. Finding the optimal new component requires a global search, which is performed by creating CN_{cand} candidate components. The number of candidates will increase linearly with the number of components C, having N_{cand} candidates per each existing component. The candidate resulting in the highest likelihood when inserted into the (previous) mixture is selected. The parameters and weight of the best candidate are then used in Step 3 instead of the truly optimal values.

The candidates for executing Step 2 are initialized as follows: the training data set X is partitioned into C disjoints data sets $\{A_c\}$, $c = 1\ldots C$, according to the posterior probabilities of individual components; the data set is Bayesian classified by the mixture components. From each A*c* number of N_{cand} candidates are initialized by picking uniformly randomly two data points x_l and x_r in A*c*. The set A*c* is then partitioned into two using the smallest distance selection with respect to x_l and x_r. The mean and covariance of these two new subsets are the parameters for two new candidates. The candidate weights are set to half of the weight of the component that produced the set A*c*. Then new x_l and x_r are drawn until N_{cand} candidates are initialized with A*c*. The partial **EM** algorithm is then used on each of the candidates. The partial **EM** differs from the **EM** and **CEM** algorithms by optimizing (updating) only one component of a mixture; it does not change any other components. In order to reduce the time complexity of the algorithm a lower bound on the log-likelihood is used instead of the true log-likelihood. The lower-bound log-likelihood is calculated with only the points in the respective set A*c*. The partial **EM** update equations are as follows:

$$w_{i,C+1} = \frac{\alpha\,\mathcal{N}(x_i,\mu,\Sigma)}{(1-\alpha)p_C\,(x) + \alpha\,\mathcal{N}(x_i,\mu,\Sigma)} \tag{23}$$

$$\alpha = \frac{1}{\mathcal{N}(A_c)} \sum\nolimits_{i\in A_c} w_{i,C+1} \tag{24}$$

$$\mu = \frac{\sum_{i\in A_c} w_{i,C+1} x_i}{\sum_{i\in A_c} w_{i,C+1}} \tag{25}$$

$$\Sigma = \frac{\sum_{i\in A_c} w_{i,C+1}(x_i-\mu)(x_i-\mu)^T}{\sum_{i\in A_c} w_{i,C+1}} \tag{26}$$

where N(A*c*) is the number of training samples in the set A*c*. These equations are much like the basic **EM** update equations in Eqs. (19) - (21). The partial **EM** iterations are stopped when the relative change in log-likelihood of the resulting $C + 1$ –component mixture drops below threshold or maximum number of iterations is reached. When the partial **EM** has converged the candidate is ready to be evaluated.

2.2.3 Figueiredo-Jain Algorithm

The Figueiredo-Jain (FJ) [8,10,12,13] algorithm tries to overcome three major weaknesses of the basic EM algorithm. The EM algorithm presented previous section requires the user to set the number of components and the number will be fixed during the estimation process. The FJ algorithm adjusts the number of components during estimation by annihilating components that are not supported by the data. This leads to the other EM failure point, the boundary of the parameter space. FJ avoids the boundary when it annihilates components that are becoming singular. FJ also allows starting with an arbitrarily large number of components, which tackles the initialization issue with the EM algorithm. The initial guesses for component means can be distributed into the whole space occupied by training samples, even setting one component for every single training sample.

The classical way to select the number of mixture components is to adopt the "model-class/model" hierarchy, where some candidate models (mixture pdf's) are computed for each model-class (number of components), and then select the "best" model. The idea behind the FJ algorithm is to abandon such hierarchy and to find the "best" overall model directly. Using the minimum message length criterion and applying it to mixture models leads to the objective function:

$$\Lambda(\theta, X) = \frac{V}{2} \sum_{c:\, \propto_c > 0} \ln\left(\frac{N\alpha_c}{12}\right) + \frac{C_{nz}}{2} \ln\frac{N}{12} + \frac{C_{nz}\,(V+1)}{2} - \ln \mathcal{L}\,(X, \theta) \tag{27}$$

Where N is the number of training points, V is the number of free parameters specifying a component, and C_{nz} is the number of components with nonzero weight in the mixture ($\propto_c > 0$). θ in the case of Gaussian mixture is the same as in (Eq. 3) The last term ln $L(X, \theta)$ is the log-likelihood of the training data given the distribution parameters (Eq. 13).

The EM algorithm can be used to minimize Eq. 27 with a fixed C_{nz} . It leads to the M-step with component weight updating formula:

$$\propto_c^{i+1} = \frac{\max\left\{0, \left(\sum_{n=1}^{N} w_{n,c}\right) - \frac{V}{2}\right\}}{\sum_{j=1}^{C} \max\left\{0, \left(\sum_{n=1}^{N} w_{n,c}\right) - \frac{V}{2}\right\}}. \tag{28}$$

This formula contains an explicit rule of annihilating components by setting their weights to zero.

The above M-steps are not suitable for the basic EM algorithm though. When initial C is high, it can happen that all weights become zero because none of the components have enough support from the data. Therefore a component-wise EM algorithm (CEM) is adopted. CEM updates the components one by one, computing the E-step (updating W) after each component update, where the basic EM updates all components "simultaneously". When a component is annihilated its probability mass is immediately redistributed strengthening the remaining components.

When CEM converges, it is not guaranteed that the minimum of $\Lambda(\theta, X)$ is found, because the annihilation rule (Eq. 28) does not take into account the decrease caused by decreasing C_{nz}. After convergence the component with the smallest weight is removed and the CEM is run again, repeating until $C_{nz} = 1$. Then the estimate with the smallest $\Lambda(\theta, X)$ is chosen. The implementation of the FJ algorithm uses a modified cost function instead of $\Lambda(\theta, X)$.

$$\Lambda'(\theta, X) = \frac{V}{2} \sum_{c: \propto_c > 0} \ln \propto_c + \frac{C_{nz}\,(V+1)}{2} \ln N - \ln \mathcal{L}\,(X, \theta). \tag{29}$$

3. Experiments Set Up

The experiments were performed using audio database extracted from video, which is encoded in raw UYVY. AVI 640 x 480, 15.00 fps with uncompressed 16bit PCM audio; mono, 32000 Hz little endian. The capturing devices for recording the video and audio data were: Allied Vision Technologies AVT marlin MF-046C 10 bit ADC, 1/2" (8mm) Progressive scan SONY IT CCD; and Shure SM58 microphone. Frequency response 50 Hz to 15000 Hz. Unidirectional (Cardiod) dynamic vocal microphones. The extracted 16 bit PCM audio files (with wav header), were sampled at 16000 Hz, mono little endian. Thirty subjects were used for the experiments in which twenty-six are males and four are females. For each subject, six multilingual (.wav files) of one minute each recording were used for each subject. The database obtained from eNTERFACE 2005 [14]. For the experts, four speech samples were used for the modeling (training); two samples were used for the subsequent validation and testing. Three sessions of the speech database were used separately. Session one was used for training the speech experts. Each expert used ten mixture client models. To find the performance, Sessions two and three were used for obtaining expert opinions of known impostor and true claims.

Performance Criteria

The basic error measure of a verification system is false rejection rate (*FRR*) and false acceptance rate (*FAR*) as defined in the following equations:

False Rejection Rate (***FRRi***): is an average of number of falsely rejected transactions. If ***n*** is a transaction and x(n) is the verification result where 1 is falsely rejected and 0 is accepted and ***N*** is the total number of transactions then the personal False Rejection Rate for user ***i*** is

$$FRR_i = \frac{1}{N}\sum_{n=1}^{N} x(n) \tag{30}$$

False Acceptance rate (***FARi***) is an average of number of falsely accepted transactions. If ***n*** is a transaction and x(n) is the verification result where 1 is a falsely accepted transaction and 0 is genuinely accepted transaction and ***N*** is the total number of transactions then the personal False Acceptance Rate for user ***i*** is

$$FAR_i = \frac{1}{N}\sum_{n=1}^{N} x(n) \tag{31}$$

Both *FRRi* and *FARi* are usually calculated as averages over an entire population in a test. If P is the size of populations then these averages are

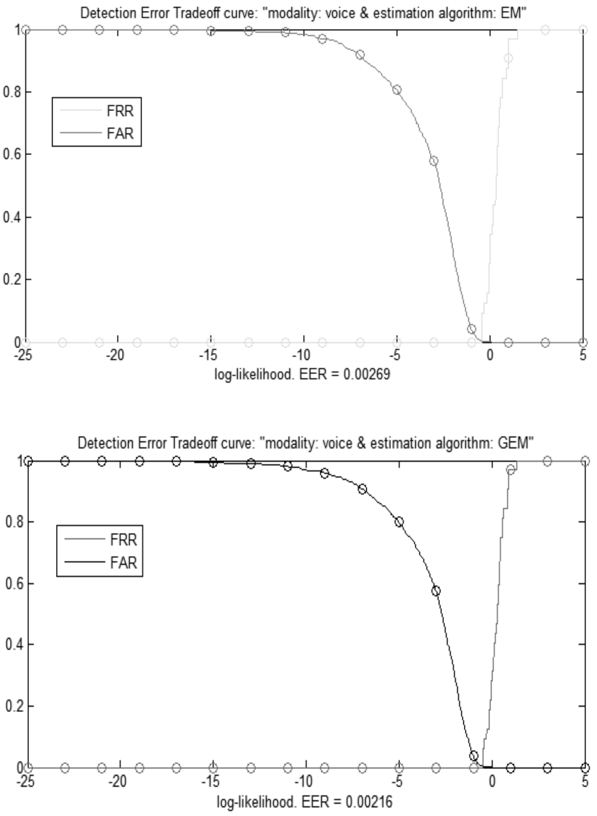
$$FRR = \frac{1}{P}\sum_{i}^{P} FRR_i \tag{32}$$

$$FAR = \frac{1}{P}\sum_{i}^{P} FAR_i \tag{33}$$

Equal Error Rate (***EER***), is an intersection where FAR and FRR are equal at an optimal threshold value. This threshold value shows where the system performs at its best.

4. Results and Discussions

As a common starting point, classifier parameters were selected to obtain performance as close as possible to *EER* on clean test data (following the standard practice in the face and speaker verification area of using *EER* as a measure of expected performance). A good decision is to choose the decision threshold such as the false accept equal to the false reject rate. In this paper it uses the Detection Error Tradeoff (DET) curve to visualize and compare the performance of the system (see Figure 3).



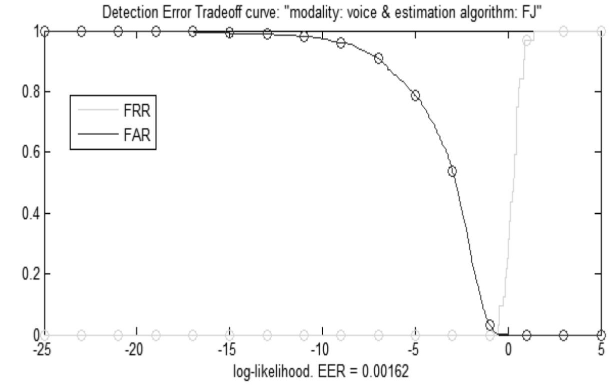


Figure 3. Detection error tradeoff curves

The (EM), (GEM) and (FJ) estimation algorithms based state-of-the art finite mixture modal (GMM) achieve a significant performance rates, EER=0.26 % for "EM", EER=0.21 % for "GEM" and EER=0.16 % for "FJ". Thus, the behavioral information scheme based speech biometrics presented a robust base with a discriminating power to the identity authentication.

5. Conclusions and Future Work

The paper has presented a human authentication method of behavioural biometrics speech information. Simulation results show that state-of-the art finite mixture modal (GMM) is quite effective in modelling the genuine and impostor score densities. The simulation results are showed a significant performance rate based EER. Hence, the behavioral information scheme based speech biometrics is robust and have a discriminating power, which can be explored for identity verification.

Furthermore it concentrated on the fusion with other modalities. Handwriting signature, dynamic face and fingerprint for the target to refine more the EER performance.

References

[1] Gleni, Sofia Petratos**,** Panagiotis (2004). *DNA Smart Card for Financial Transactions,* The ACM Student Magazine.

[2] Chetty, Girija., Wagner, Michael (2006). Audio-Visual Multimodal Fusion for Biometric Person Authentication and Liveness Verification *In*: NICTA-HCSNet Multimodal UserInteraction Workshop (MMUI2005), Michael Australian Computer Society, Inc. Sydney, Australia.

[3] Poh, Norman., Bengio, Samy (2004). Database, Protocol and Tools for Evaluating Score-Level Fusion Algorithms in Biometric Authentication, IDIAP RR 04-44, August, a IDIAP, CP 592, 1920 Martigny, Switzerland.

[4] Sanderson, Conrad., Bengio, Samy., Bourlard, Herve., Mariéthoz, Johnny., Collobert, Ronan., BenZeghiba, Fabien., Cardinaux, Mohamed F., Marcel, S´ebastien (2005). Speech & Face based biometric authentication at DIAP", Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP). Rue du Simplon 4, CH-1920 Martigny, Switzerland.

[5] Vielhauer, Claus., Schimke, Sascha., Thanassis, Valsamakis., Stylianou, Yannis (2005). Fusion Strategies for Speech and Handwriting Modalities in HCI, Multimedia on Mobile Devices, edited by Reiner Creutzburg, Jarmo H. Takala, *In*: Proc. of SPIE-IS&T Electronic Imaging.

[6] Mølgaard, Lasse L., Jørgensen, Kasper W. (2005). Speaker Recognition: Special Course, *In*: IMM _DTU December 14.

[7] Davis, S., Mermelstein, P (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Transactions on Acoustics, Speech and Signal Processing*, (4) 357–366.

[8] Paalanen, Pekka (2004). Bayesian classification using gaussian mixcute model and EM estimation: implementation and comparisons, Information Technology Project, Lappeenranta, June 23, http://www.it.lut.fi/project/gmmbayes/

[9] Pham Minh, Tri. On estimating the parameters of Gaussian mixtures using EM, School of Computer Engineering, Nanyang Technological University

[10] Paalanen, P., Kamarainen, J.-K Ilonen, J. Kälviäinen, H. (2005). Feature Representation and Discrimination Based on Gaussian Mixture Model Probability Densities: Practices and Algorithms", Department of Information Technology, Lappeenranta University of Technology, P.O.Box 20, FI-53851 Lappeenranta, Finland.

[11] Li, Lei., Ma, Jinwen (2008). A BYY Split-and-Merge EM Algorithm for Gaussian Mixture Learning, *In*: F. Sun et al. (Eds.): ISNN 2008, Part I, LNCS 5263, p. 600–609, 2008. Springer-Verlag Berlin Heidelberg.

[12] Verbeek, J.J., Vlassis, N. Krose, B. (2003). Efficient Greedy Learning of Gaussian Mixture Models, Informatics Institute, University of Amsterdam- Kruislaan 403, *Neural Computation* 15 (2) 469-485.

[13] Jan, R.J., Nunnink Jakob J., Verbeek Nikos Vlassis. Accelerated Greedy Mixture Learning. , Informatics Institute, Faculty of Science, University of Amsterdam Kruislaan 403, 1098 SJ Amsterdam, The Netherlands.

[14] Stylianou, Yannis., Pantazis, Yannis., Calderero, Felipe., Larroy, Pedro., Severin, Francois., Schimke, Sascha., Bonal, Rolando., Matta, Federico., Valsamakis, Athanasios (2005). GMM-Based Multimodal Biometric Verification, eNTERFACE 2005 The summer Workshop on Multimodal Interfaces July 18th – August 12th, Facultè Polytechnique de Mons, Belgium.

[15] Reynolds, D.A. (1994). Experimental Evaluation of Features for Robust Speaker Identification, *IEEE Trans. Speech and Audio Processing* 2 (4) 639-643.

[16] Fortuna, J., Malegaonkar, A., Ariyaeeinia, A., Sivakumaran, P. (2005). On the Use of decoupled and adapted Gaussian mizture models for openset speaker identification, University of Hertfordshire, Hatfield, UK, Canon Research Centre Europe Ltd., Bracknell, UK. The Third COST 275 Workshop, Biometrics on the Internet. University of Hertfordshire Hatfield, UK 27-28 October.

[17] Sànchew-Soto., Eduardo Blouet., Raphaél., Sigelle, Marc., Chollet, Gérard (2004). Model Adaptation for Speaker Verification using Conditional probability tables in Bayesian Networks", Ecole Nationale Supérieure des Télécommunications – Département de Traitement de Signal et des Images. LTCI/CNRS URA 820. 46 rue Barrault 75634 Paris Cedex 13 France. 2nd COST 275 Workshop – Biometrics on the Internet Vigo, 25-26 March.

[18] Cohen, Arnon., Zigel, Yaniv (2002). On feature selection for speaker verification, Electrical and Computer Engineering Department, Ben-Gurion University, Beer-Sheva, Israel. COST 275 Workshop – The Advent Biometrics on the Internet Rome, Italy November 7-8.

[19] Arcienega, Mijail., Drygajlo, Andrzej (2002). Spectral envelope features for speaker verification, Signal Processing Institute Swiss Federal Institute of Technology, Lausanne. COST 275 Workshop – The Advent Biometrics on the Internet Rome, Italy November 7-8.

[20] Matrouf, Driss., Bonastre, Jean-Franc¸ois., Costa, Jean-Pierre (2005). Effect of impostor speech transformation on automatic speaker recognition, LIA, Universit´e d'Avignon Agroparc, BP 1228- 84911 Avignon CEDEX 9, France. The Third COST 275 Workshop, Biometrics on the Internet. University of Hertfordshire Hatfield, UK 27-28 October.