Maximum likelihood Linear Programming Data Fusion for Speaker Recognition

Enric Monte-Moreno (1), Mohamed Chetouani (2), Marcos Faundez-Zanuy (3), Jordi Solé-Casals (4)

(1) TALP Research center, UPC Barcelona, Spain (2) Université Pierre et Marie Curie-Paris 6, France, (3) Escola Univesitària Politècnica de Mataró, UPC Barcelona, Spain, (4) Universitat de Vic, Barcelona, Spain.

ABSTRACT

Biometric system performance can be improved by means of data fusion. Several kinds of information can be fused in order to obtain a more accurate classification (identification or verification) of an input sample. In this paper we present a method for computing the weights in a weighted sum fusion for score combinations, by means of a likelihood model. The maximum likelihood estimation is set as a linear programming problem. The scores are derived from a GMM classifier working on a different feature extractor. Our experimental results assesed the robustness of the system in front a changes on time (different sessions) and robustness in front a change of microphone. The improvements obtained were significantly better (error bars of two standard deviations) than a uniform weighted sum or a uniform weighted product or the best single classifier. The proposed method scales computationaly with the number of scores to be fussioned as the simplex method for linear programming.

1. INTRODUCTION

Biometric recognition (Faundez-Zanuy, 2006) offers a promising approach for security applications, with some advantages over the classical methods, which depend on something you have (key, card, etc.), or something you know (password, PIN, etc.). A nice property of biometric traits is that they are based on something you are or something you do, so you do not need to remember anything neither to hold any token.

On the other hand, they have an important drawback, because if a person's biometric data is stolen, it is not possible to replace it (Faundez-Zanuy, 2004). Probably, these drawbacks have slowed down the spread of use of biometric recognition (Faundez-Zanuy, 2005b). For those applications with a human supervisor (such as border entrance control), this can be a minor problem, because the operator can check if the presented biometric trait is original or fake. However, for remote applications such as internet, some kind of liveliness detection and anti-replay attack mechanisms should be provided. Fortunately, speech offers a richer and wider range of possibilities when compared with other biometric traits, such as fingerprint, iris, hand geometry, face, etc. For instance, you can use a text-dependent system (Faundez-Zanuy and Monte-Moreno, 2005) and to ask the user for a specific speech sentence. Speaker recognition does not offer the same robustness and precision than other biometric traits such as fingerprint and iris. However, strong efforts are done to enhance the performance, due to its particular set of characteristics that can permit to manage some vulnerability attacks.

This paper is organized as follows: section two describes the different data levels for fusion with special emphasis on the score level. Some new strategies are presented for data fusion. Section three is devoted to the experimental results, and section four summarizes the main conclusions.

2. DATA FUSION

2.1 Introduction

Given a biometric system, such as that depicted in figure 1, four main data fusion levels can be defined: sensor, feature, score (also known as opinion) and decision. The description of these levels is beyond the scope of this paper and can be found in (Faundez-Zanuy, 2005a).

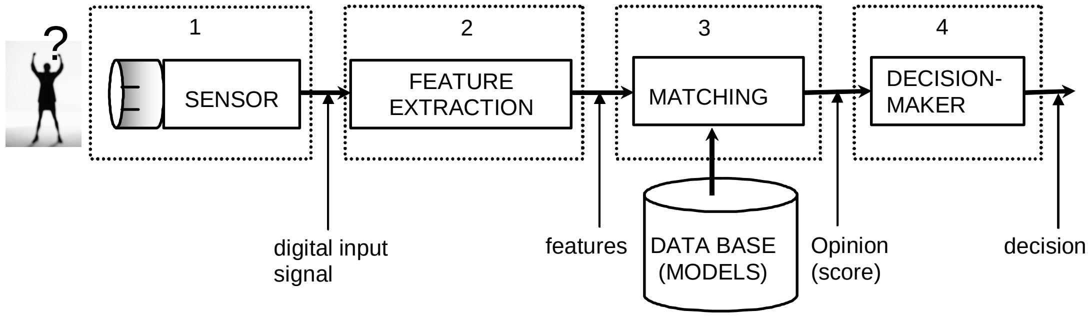

Figure 1: General scheme of a biometric system

In this paper we will focus on the score level. This kind of fusion is also known as confidence level. Given a set of classifiers (matchers), it consists of the combination of the scores provided by each matcher. The matcher just provides a distance measure or a similarity measure between the input features and the models stored on the database.
It is possible to combine several classifiers working with the same biometric characteristic (unimodal systems) or to combine different ones. In our case, it will be a unimodal combination, where both classifiers share the same input signal, as depicted in figure 2. This scheme can be easily generalized for more than two matchers.

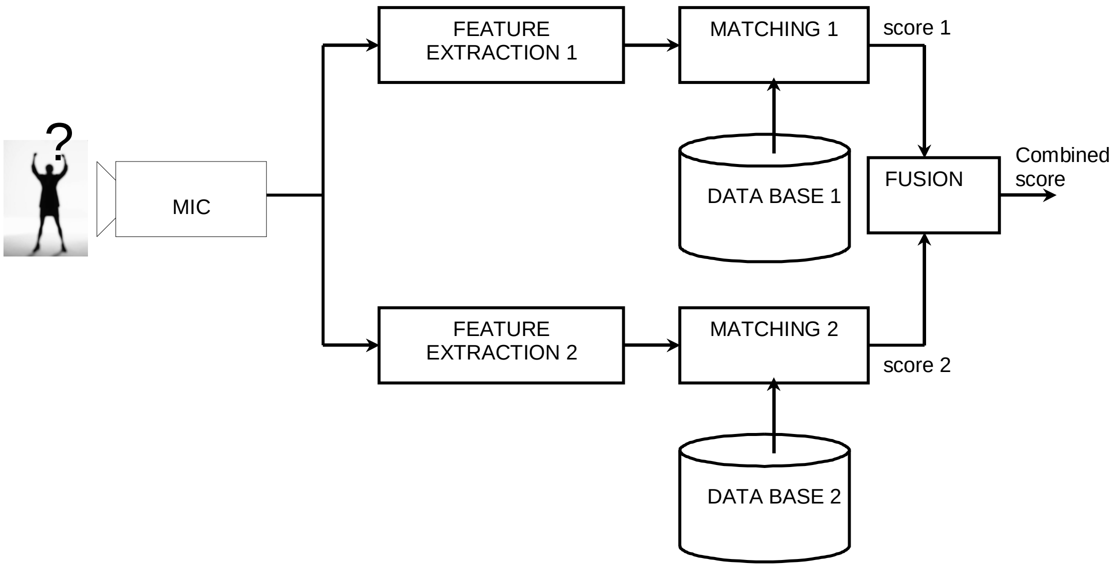

Figure 2: General scheme for data fusion at score level.

2.2 Combination strategies

The score combination schemes for a given speaker can be done in several ways (see kuncheva 2004). The most natural strategies for combining different scores, might be:

1) Weighted sum: $O_s = \sum_{j=1}^{N} h_j o_{js}$

2) Weighted product: $O_s = \prod_{j=1}^{N} \left(o_{js}\right)^{h_j}$

In this paper we propose a fusion method, where the scores will be interpreted as probabilities of an observation, given a model. In this paper for each observation we will have a vector of N-scores, which will be the probability of the identitiy of a speaker for a set of N classifiers. The global likelihood function will be the product of the all the probabilities of the observations of all speakers where each score will be weighted by a factor h_j that will be specific for a given score. The likelihood function, of these probabilities can be understood as either a weighted product of probabilities, or a weighted sum of logarithms of probabilities.

The strategy for computing the h_j parameters that weight the different scores can be done by several methods, the first and most simple might be the brute force method, which would consist on exploring the space of possible recognition rates for all possible combinations of a set of discrete values of the weighting parameters. The problem with this method is that it scales exponentially with the number of scores, and therefore is it only has sense for a small values of the number of scores to be merged (i.e. N=2,3). Another possibility might be the use of a least squares method for the estimation of the weighting parameters, without considering a likelihood model, i.e. considering the observations as a probability of an observation given a model. The use of a least squares method will assign to positive examples the target value $O^{\text{target}}{}_s = +1$, and negative examples target $O^{\text{target}}{}_s = -1$. The advantage of a least squares estimation is that it might take into account the possible correlations (positive or negative) between scores. This method was taken into consideration at the begining of the project, but had several problems: a- the introduction of restrictions on the set of parameters h_j, was artificial, so the set of equations had to be solved as a nonlinear convex optimization problem (Boyd and Vandenberghe, 2004), b- the natural way of setting the least squares problem was as a discriminative estimation (i.e positive and negative examples), which gave rise a a inconsistent system of equations[1]. The use of a discriminative model was was discarded because the set of equations to be solved by the least squares method $Ah = b$ (where A is a matrix data, and b, is the target vector, which's values are ± 1) was inconsistent, probably due to the fact that the use of negative examples made that a fraction of similar items of the training data were assigned to different labels. The problem of identifying the subset of the training data that yielded a consistent set of equations was not tried, because of the combinatorial nature of the problem. Note that even the use of suboptimal methods for estimating subsets such as the forward selection (see Bishop 1995) yields a quadratical cost with the number of examples which makes the problem computationally unfeasible.

An alternative to the use of negative examples is the use a likelihood model, which solves a different kind of problem. I.e. the computing the set of parameters that maximizes the likelihood of the combined set of scores for all the speakers simultaneously. Note that only positive samples are used.

On the other hand, the use of a likelihood model, with the introduction of restrictions on the weights h_j, gave naturaly a set of equations that were equivalent to a linear programming problem.

[1] The inconsistency test is given by the simplex algorithm (see Bertsimas 1997)

The fusion process will be done by means of the following model,

$$L(x) = \prod_{s=1}^{S} \prod_{i=1}^{M} \prod_{j=1}^{N} P_{s,j}^{h_j}\left(x_i\right) \tag{1}$$

where $P_{s,j}^{h_j}(x)$ is the probability of the sample x_i given the model for the speaker *s*, and the parameterization *j* The weighting parameter h_j weights the contribution of the parameterization *j* to the global likelihood. This parameter is specific of the parameterization and independent of the speaker. The number parameterized samples of speaker *s* is *M*. The total number of speakers is denoted by S.

The goal is to find the values h_j that maximize the likelihood (1) in a geometrical simplex, i.e. $Simplex = \left\{ \left(h_1 \ \cdots \ h_N\right) \in \mathbb{R}^N \mid \sum_{j=1}^{N} h_j = 1 \, and \, h_j \geq 0, \forall j \right\}$. A simplex constraint was selected, in order to restrict the possible values of h_j, because of the fact that the estimates are found by maximizing the likelihood function (1), which can be unbounded for negative values of h_j, or can give rounding errors for $h_j >> 1$. Another reason for selecting a solution in a simplex, is that the optimization algorithm will allocate a limited 'budget' of probability between the different scores, and therefore the scores that contribute marginally to the correct fusion will be given low values of h_j (notice that $h_j = 0$ makes irrelevant a parameterization), while the rest of the probability budget will be allocated to the parameterizations that contribute most to the correct fusion.

The function to be maximized (1) can be set for a given speaker *s* as a log-likelihood function,

$$\begin{aligned} & h = \arg\max \left(\sum_{i=1}^{M} \sum_{j=1}^{N} h_j \ln\left(P_{s,j}\left(x_i\right)\right) \right) \\ & subjet\ to \quad \begin{cases} \sum_{j=1}^{N} h_j = 1 \\ h_j \geq 0 \end{cases} \end{aligned} \tag{2}$$

Our objective is to find a the vector *h* that maximizes simulatenously the likelihood for each speaker. We decided to express the optimization problem with a restriction on each speaker in order to control a common margin, so that each speaker will have a likelihood at least as high as the value of a positive threshold. Notice that if the objective function in (2) had a sum for all speakers, we would not be able to control the likelihood of the worst speaker. Therefore we introduced a new variable which is the common positive threshold for the likelihoods of all speakers, which we will denote as δ, and the result of the optimization process will be the value of δ plus the values of h that are compatible with the restriccions.

This problem can be expressed in a convex optimization framework (Boyd and Vandenberghe, 2004) as:

$$
\begin{aligned}
&\max_{\delta} \quad DeltaWeight \cdot \delta \\
&\quad subjet\ to \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3)\\
&\quad \begin{cases} Ah \geq \delta e \\ \sum_{j=1}^{N} h_j = 1 \\ h_j \geq 0 \\ \delta \geq 0 \end{cases}
\end{aligned}
$$

Where A is a matriz of $(M\times S)\times N$ with the following structure:
$A = \begin{bmatrix} A^1 & \cdots & A^s & \cdots & A^S \end{bmatrix}^T$
and each A^s is a submatrix of $(M\times N)$ composed by $a^s{}_{i,j} = \ln\left(P_{s,j}(x_i)\right)$. The optimization variable is δ and e is a column vector of dimension $(M\times S)$. The restrictions on the function to be maximized (2) is that simultaneously for all speakers, the weighted scores of every utterance of speaker s, $\sum_{j=1}^{N} h_j \ln\left(P_{s,j}(x_i)\right)$ will have a higher value than the variable to be maximized δ. This variable is weighted in the objective function by a parameter that we will denote as Delta weight, which can be seen as a scale factor over the log probabilities, which will work as a trade-off in the simplex $\sum_{j=1}^{N} h_j = 1$ generated by h_j. Low values of the delta weight will give solutions near the baricenter (center of mass) of the simplex $\sum_{j=1}^{N} h_j = 1$, while high values will give solutions near a vertex of the simplex. This value might be seen as a prior over the h_j set of values, in the sense that low values of the Delta weight, will yield a solution more or less uniform, while high values of the Delta weight will give a sparse solution allocating most of the probability mass to a reduced number of scores. As will be seen in section 3.6, there is a trade-off in the performance of the classifier, which can be controlled by means of this parameter.
This optimization problem is solved by means of the simplex algorithm[2] (Bertsimas and Tsitsiklis, 1997).

The problem (3) can be expressed as a standard linear programming problem;

$$
\begin{aligned}
&\min_{x} \quad f^T x \\
&\quad subjet\ to \qquad\qquad\qquad\qquad (4)\\
&\quad \begin{cases} Ax \leq b \\ A_{eq}x = b_{eq} \\ lb \leq x \leq ub \end{cases}
\end{aligned}
$$

where A is the matrix of log probabilities, which was defined in (3) and $f, x, b, b_{eq}, lb, ub\ and\ A_{eq}$ are vectors defined as:

[2] The simplex algorithm for solving the linear programming problem, should not be confused with the geometrical simplex, which is a constraint on the parameters to be estimated.

$$
\begin{cases}
f = \left[\underbrace{0,\cdots 0}_{N},\ DeltaWeight \right]^T \\
x = \left[h_1, h_2, \cdots h_N, -\delta \right]^T \\
b = \left[0 \cdots 0 \ \right]^T \\
b_{eq} = [1]^T \\
lb = [0 \cdots 0]^T \\
ub = [1 \cdots 1]^T \\
A_{eq} = \left[\underbrace{1 \cdots 1}_{N}, 0 \right]^T
\end{cases} \tag{5}
$$

The method we propose has several computational advantages, perhaps the most interesting is that the *average* case running time for the simplex algorithm for linear programming. Although some examples can be contructed where the simplex algorithm can take an exponential time with the number of constraints, the mean time, is a polinomial of the number of contraints, which makes the solution quite inexpensive from the computational point of view (Bertsimas 1997).

3. EXPERIMENTAL RESULTS

3.1 Database

The Gaudi database (Ortega et al., 2000; Satue and Faundez-Zanuy 1999) was originally designed in order to measure the performances under different controlled conditions: language, interval session, microphone. The corpus is composed by:

- 49 speakers.
- 4 sessions with different tasks: isolated numbers, connected numbers, read text, conversational speech, etc. ...).
- For each session, the utterances have been acquired in two languages (Catalan and Spanish) and simultaneously with different microphones as described in table 1.

Table 1 The microphones used for the Gaudi database.

MIC1	SONY ECM 66B	lavalier unidirectional electret (≈10 cm from the speaker)
MIC2	AKG D40S	dynamic cardioid (≈30 cm from the speaker)
MIC3	AKG C420	head-mounted (low-cost microphone)

In this contribution, the training protocol consists of using one reading text of an average duration of one minute (using session 1 and MIC1). Concerning the tests, we use 5 phonologically balanced utterances (Spanish) identical for all the speakers through the scenarios M3 to M6. We focus on the third first sessions with different microphones (cf. table 2) The number of tests for genuine users is 49×5=245 for each session and the average score is estimated under 49×5×6=1470 tests.

Table 2. Different sessions and microphones notation.

Scenario	Session	Microphone
M1	1	MIC1
M2	1	MIC2
M3	2	MIC1
M4	2	MIC2
M5	3	MIC1
M6	3	MIC3

The speech signal has been down-sampled to 8 kHz, pre-emphasized by a first order filter whose transfer function is $H(z)=1-0.95z^{-1}$ and normalized between -1,+1 (for cumulant estimation). A 30ms Hamming window is used, and the overlapping between adjacent frames is 2/3. A parameterized vector of order 16 was computed for each feature extraction method.

3.2 Feature extraction

Sate-of-the-art feature extraction methods are based on the MFCC (Mel Frequency Cepstral Coding) or the LPCC (Linear Predictive Cesptral Coding). These short-term features are currently used in GMM based speaker recognition systems. Alternative features have been investigated resulting on different approaches. The first ones consist of the development of short-term features (as LPCC or MFCC) such as the use of signal decomposition methods (Wavelet, Independent Component Analysis). Other techniques aim to exploit other levels of representation such as phonetic, prosodic, idiolectal, dialogic or semantic (Faundez-Zanuy and Monte-Moreno, 2005). These features are extracted from long-term physical traits and are usually fused with the traditional spectral features (short-terms).

In this contribution, we propose to evaluate additional short-term features that can also be combined with the MFCC/LPCC ones. These features are extracted from the LP-residue.

3.2.1 Feature Extraction from the Residue

Speech signals are assumed to result from the excitation of the vocal tract according to the source-filter model. Following the LPC analysis framework, the vocal tract is associated to the filter (LPC coefficients) and the excitation to the residual signal. The LP analysis consists of the estimation of LPC coefficients by minimizing the prediction error. The predicted sample results from a linear combination of the *p* past samples (Atal and Hanauer, 1971):

$$\hat{s}(n) = -\sum_{k=1}^{p} a_k s(n-k) \tag{6}$$

The LPC coefficients a_k are related to the vocal tract and should also partly capture speaker-dependent information. Indeed, derived features from these coefficients, namely the Linear Predictive Cepstral Coding (LPCC), are intensely used in speaker recognition tasks. The parameter *p* (filter's order) plays a major role for instance in speech recognition tasks best scores are obtained with 12th order whereas in speaker recognition the most used order is 16.

Within the traditional LP analysis, the residual is obtained by the error between the current and the predicted samples:

$$r(n) = s(n) - \hat{s}(n) \qquad (7)$$

Theoretically, the residual is uncorrelated to the speech signal and it is related to the excitation which is speaker-dependent. These features are known as source features. However, recent works on non-linear speech processing have shown that the source-filter model is not suitable for the speech production modelling (Faundez et al. 2002; Kubin, 1995). Different phenomena occur during the production, that are non-linear and chaotic. From these investigations on non-linear processing, one can assume that there is a dependency between the speech signal and the residual.

Several investigations have been carried out for the use of this residual for the improvement of speaker recognition systems (Thevenaz and Hügli., 1995, Faundez and Rodriguez, 1998; Mary et al.20 04; Yegnanaraya, 2001; Mahadeva et al,. 2006; Zheng et al. 2006). Thevenaz and Hügli (Thevenaz P. and Hügli, 1995) exploit the theoretical orthogonality between two models respectively the filter (i.e. the LPC coefficients) and the residue. Their results confirm the complimentary of these representations for speaker verification. Neural networks have been also tested for the characterisation of the LP residual (Mary et al,. 2004). In (Mahadeva et al,. 2006), Auto-associative neural networks are used for the characterisation of the linear residue. They show that speaker recognition systems can reach efficient rates by using only residual features.

In this contribution, we propose to exploit the fact that the residue conveys all information that are not modelled by the LPC filter (cf. equation 7). These informations are modelled by two techniques: temporal and frequential. The first approach attempts to model the residual signal by an Auto-Regressive (AR) model while the second one is based on a filter bank based model.

Temporal approach:

The temporal approach is based on an Auto-Regressive (AR) model of the LP-residue:

$$\hat{r}(n) = -\sum_{k=1}^{\rho} \alpha_k r(n-k) \qquad (8)$$

Where r and ρ respectively represent the LP-residue and the filter's order.

Auto-regressive coefficients (i.e. LPC features) are not directly used in speech applications. LPCC features obtained from the LPC by a cepstral transformation are prefered due to their decorellation properties suitable for diagonal matrices based models (GMMs). The α_k coefficients are transformed on cepstral ones γ_k similarly to the LPC-LPCC transformation. The obtained cepstral features are termed the R-SOS-LPCC since they are obtained from a cepstral transformation of an AR modelling of the LPC residue.

Frequential approach:
Contrary to the previous approach, in this section, we describe a frequential processing of the residual signal r(n). This approach was originally proposed by Hayakawa and al. (Hayakawa et al, 1997} and called by them the Power Difference of Spectra in Subband (PDSS). They tested it on a speaker identification problem, the R-PDSS features gave a rate of 66.9% and the combination with LPCC features gave 99% (99.8% for the LPCC alone).

The R-PDSS features are obtained by the following steps :

- Calculate the LP-residual r.
- Fast Fourier Transform of the residual using zero padding in order to increase the frequency resolution: $S=|fft(residue)|^2$.
- Group the power spectrum into M sub-bands.
- Calculate the ratio of the geometric to the arithmetic mean of the power spectrum of the i^{th} sub-band and subtract it to 1:

$$R-PDSS(i)=1-\frac{\left(\prod_{k=L_i}^{H_i} S(k)\right)^{\frac{1}{N_i}}}{\frac{1}{N_i}\sum_{k=L_i}^{H_i} S(k)} \qquad (9)$$

Where $N_i = H_i - L_i + 1$ is the number of sample number of frequency points in the i^{th} sub-band. L_i and H_i are respectively the lower and upper frequency limits of the i^{th} sub-band. The same bandwidth is used for all the sub-bands.

Cepstrum analysis of the residual has been also investigated in speech recognition (He et al., 1996): filter bank analysis of the one-sided auto-correlation of the residual *r* plus a cepstral transformation. The obtained features named as RCEP (Residual Cepstrum) present some linguistic information and in combination to the LPCC improves the recognition rates.

3.3 Feature Linearization

Communications channel can be modeled as a linear filter, in a simplest case, or as a Wiener system (linear filter followed by a nonlinear invertible function). Many research have been done in the identification and/or the inversion of linear and nonlinear systems. These works assume that both the input and the output of the distortion are available (Prakriya and Hatzinakos, 1985); they are based on higher-order input/output cross-correlation (Bellings and Fakhouri, 1978) bispectrum estimation (Nikias and Petropulu, 1993; Nikias and Raghuveer, 1987) or on the application of the Bussgang and Prices theorems (Boer, 1976; Jacoviti et al., 1987) for nonlinear systems with Gaussian inputs.

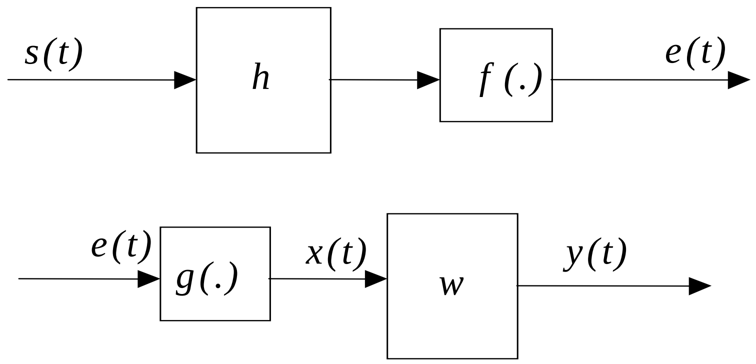

Figure 3: The unknown nonlinear convolution system (top) and the proposed inversion structure (bottom)

However, in real world situations one often does not have access to the input. In this case, blind identification becomes the only way to solve the problem.
One of the main sources of degradation in speaker recognition is the mismatch between training and testing conditions. This is due because in most of the situations we can not control the channel effects over the speech signal. It means that the parameters extracted in the recognition stage can be modified for the channel effects and can cause that system fails to recognize an authorized speaker.
In order to minimize the channel effects, we try to homogenize the channel effects by means of a linearization effect. Other strategies can be found in (Sole-Casals and Faundez-Zanuy, 2006).
We use a homogenization method inspired on recent advances in source separation of nonlinear mixtures (see (Sole-Casals et al, 2002; Taleb and Jutten, 1999; Taleb et al., 2001; for details) . Based on the inversion of Wiener systems or Post-Nonlinear mixtures in BSS/ICA context, we propose to Gaussianize the speech signal before to extract the parameters as is done in (Sole-Casals et al, 2005).

3.3.1 Cumulative density function

The simplest approach for computing gi is based on the property of the cumulative density function (cdf). Consider the random variable *E*, and denote its cdf $F_E = \Pr(E < u)$, where $\Pr(\;)$ denotes the probability. The random variable $U = F_E(E)$ is then uniformly distributed in [0, 1]. Denoting by $\Phi(u)$ the Gaussian cdf, which transforms a unit variance Gaussian variable into a uniform random variable in [0, 1], it is clear that $\Phi^{-1}(u)$ is a unit variance Gaussian random variable. Then, a simple Gaussianization procedure (see figure 2) is to apply this direct method, provided we have the function $\Phi^{-1}(\;)$, by using the following nonlinear mapping:

$$g = \Phi^{-1} \circ F_E \qquad (10)$$

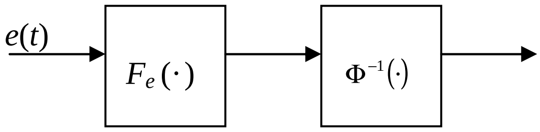

Figure 4: The system Gaussianization for a speech signal *e*(*t*). The first block consists in estimating the cumulative density function (cdf) of the observed signal and the second block is the inverse of the Gaussian cdf.

3.3.2 Maximization of Shannon entropy

Let $p_z(u)$ denote the probability density function of Z, the Shannon entropy of the unit variance random variable Z, defined by:

$$H(Z) = \int -\log(p_z(u))p_z(u)du \tag{11}$$

is maximum if Z is Gaussian (Cover and Thomas, 1991). Then, another Gaussianization method can be obtained so that $H(Z)$ is maximum (under the constraint of unit variance).

3.3.3 Algorithms

Using the previous results, one can propose two algorithms for the linearization (Gaussianization) of the speech signal. The first algorithm is based on formula (10). The Matlab code is very simple and very fast. A second algorithm, based on (11), consists of adjusting a nonlinear mapping g so that the Shannon's entropy of $Z = g(E)$ is maximum under the constraint $Ez^2 = 1$. Although the second idea is still quite simple, it leads to an algorithm which is much more complicated and requires much iterations before converging to an acceptable solution. On the contrary, the algorithm based on (10) provides an analytical solution without any iterations. In the following, we only consider this fast algorithm.

3.4 Classification

The classification system is based on standard Gaussian Mixture Models (GMMs) (Reynols and Rose, 1995). A Gaussian mixture density is a weighted sum of *K* component densities given by:

$$P(x/\lambda) = \sum_{k=1}^{K} \omega_k g_{(\mu_k,\Sigma_k)}(x) \tag{12}$$

Where x is a d-dimensional vector, $g_{(\mu,\Sigma)}(x)$ are the component denisities and ω_k the mixture weights. Each component density is a d-variate Gaussian function:

$$g_{(\mu,\Sigma)}(x) = \frac{1}{(2\pi)^{d/2}\sqrt{\det(\Sigma)}} e^{-1/2(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{13}$$

With mean vector μ_k and covariance matrix Σ_k. The mixture weights ω_k satisfy the following constraint:

$$\sum_{k=1}^{K} \omega_k = 1 \tag{14}$$

The Gaussian Mixture Model is defined by the mean vectors, covariance matrices and mixture weights. The set of parameters is grouped and represented by:

$\lambda = (w_k, \mu_k, \Sigma_k) \quad k=1\ldots K.$

Each speaker is modelled by a GMM with 32 mixtures and diagonal covariance matrices.

3.5 Normalization of the scores.

In the case of fusion it is usual to introduce a normalization of the scores, so that the fusion is done on adimensional units, which behave in a statisticaly similar fashion. In our case, there was no need of normalizing the distance measures. The set of classifiers to be merged were homogeneous, and the only difference was due to the parameterization. The margin of variation of the measures was similar, as can be seen in figure 5.

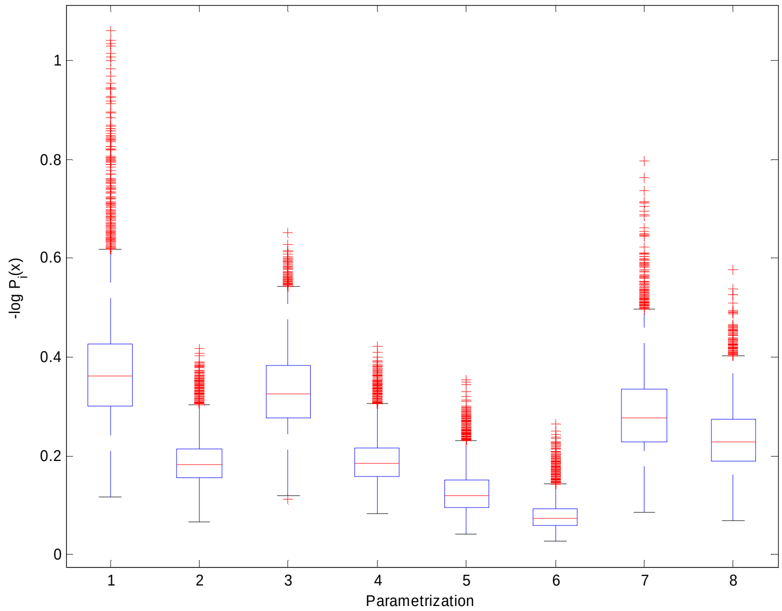

Figure 5: Box plot of the distances of all the utterances, ordered by parametrization**.** The parametrization titles are shown in table 3.

Table 3. Coding of the parametrization

1	LPCC
2	LPCC_linearization
3	MFCC
4	MFCC_linearization
5	PDSS
6	PDSS_linearization
7	SosResidualLPCC
8	SosResidualLPCC_linearization

3.6. Results of the linear programming fusion

We have compared the fusion method based on linear programing with the computation of the mean value of the classification results of each parameterization. The fusion method consisted on the linear combination of the outputs of the classifiers on two cases: a- The h_j were set to 1/N (i.e. uniform weighting of each score) b- The The h_j computed by means of the linear programming method.

Another possibility was to compare the results of the fusion with the parameterization (or a subset of parametrizations) that gave the best results. The results did not show a consistent behaviour. Some parametrizations were better in the sense of robustness in front of a change of session, but had a bad performance when the microphone was changed, and others degraded with a change of a microphone. In any case the fusion method based on the linear programming method consistently improved over the best method alone. For comparisons purposes we will present the results of the two different fusion methods with the recogntion results of the parametrization that globally gave the best results, i.e. MFCC.

The experiments were designed in order to see the robustness of the fusion method with respect to either a change of session or a change of microphone. As reference we took scenario M1 (see section 3.1), which consisted of training with session 1 and microphone 1 and with four of the five phrases and recognizing with the left out phrase. This was repeated for all the phrases, and the results are shown in figure 6.

In all figures, the error bars represent two standard deviations, i.e. a confidence interval of 95%.

Notice that using the linear programming model improves significantly the recognition rate for the different test sentences in the reference set up.

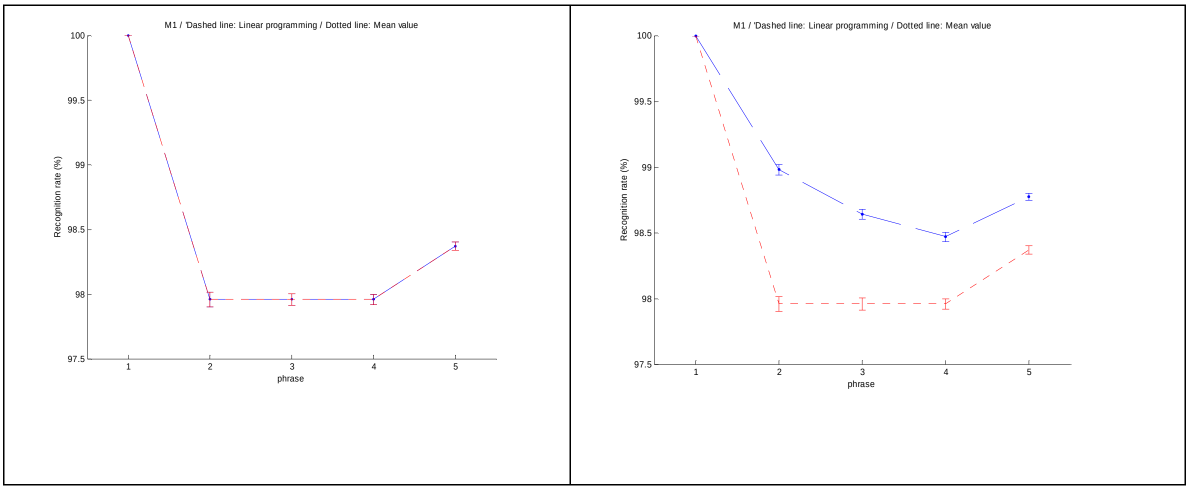

Figure 6: Reference results M1, for Delta weight : 1 (left),10 (right)..

The Delta weight, as explained in section 2.2 controls the flatness of the weighting vector. The experiments showed that low values of the delta weight gave a near uniform distribution of the weights, while high values, selected the weights that can be understood as the most relevant. Notice that as the training was not discriminative, the parmetrization with the highest values h_j should not be taken as the most discriminative, but as the ones that contribute more to the likelihood of the data given the model. Figure 7 shows the values of the h_j for values of the Delta weight ={1,10}.

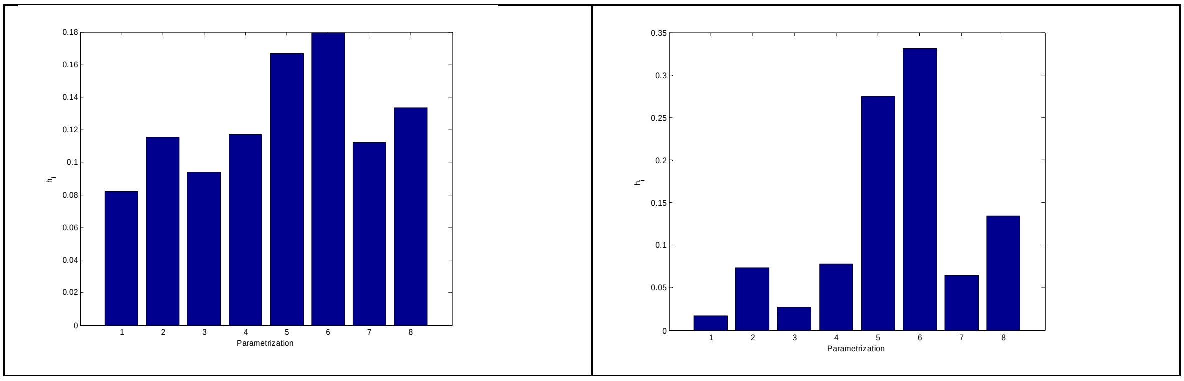

Figure 7: Values of the h_j for different values of the Delta weight: 1 (left),10 (right).

The first experiment of interest is the rubustness of the method with respect to a change in the date of the recording (i.e. the session), but without changing the microphones, which correspond to scenario M3 and M5. We computed the weighting parameters h_j on scenario M1, and tested with M3 and M5. In case of M3, which corresponds to session 2, the sentences 4 and 5 were not distinguished, and in session M5, the use of a high value of Delta weight, yields a significative improvement. See figures 8 and 9.

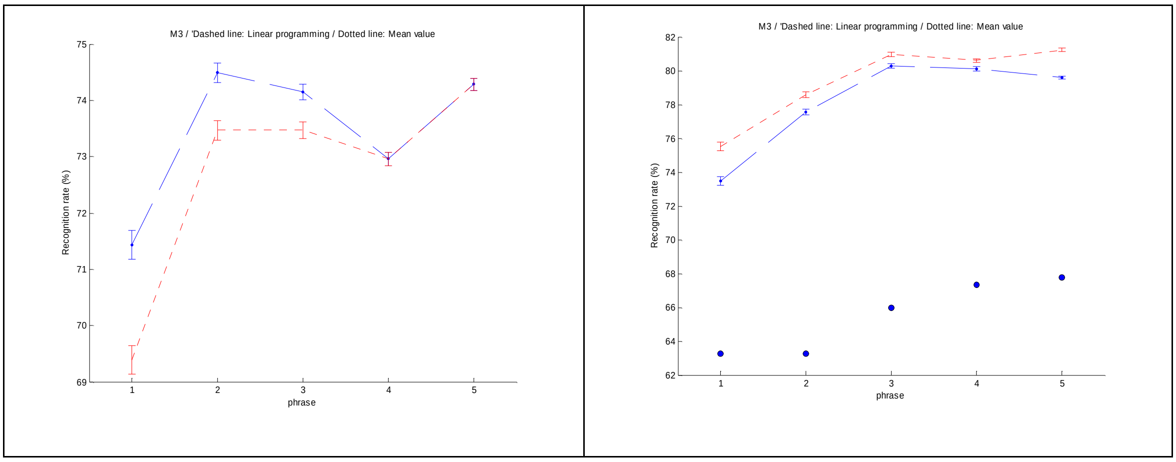

Figure 8: Robustness of the method with respect to a change in the date of the recording. Setting M3 for Delta weights: 1 (left),10 (right).

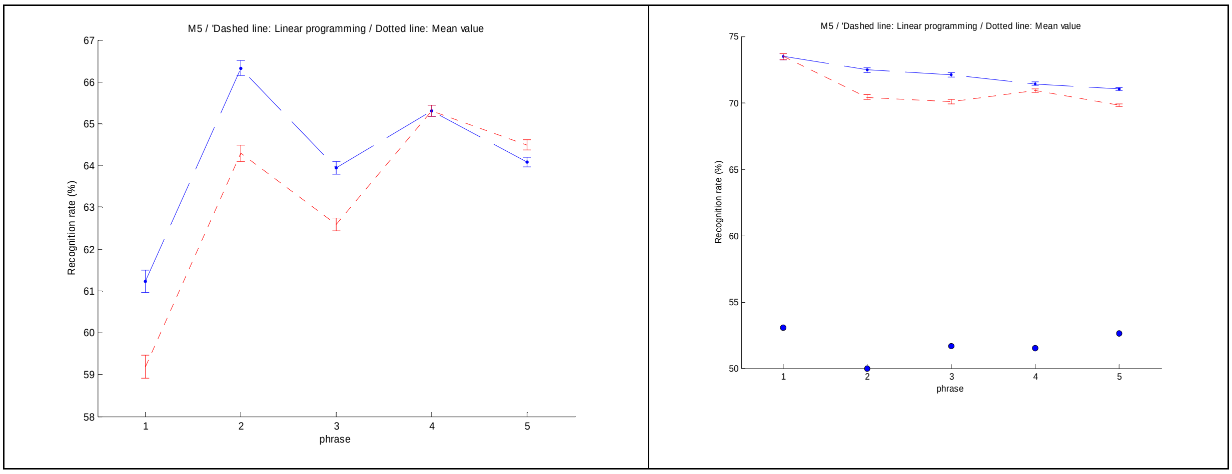

Figure 9: Robustness of the method with respect to a change in the date of the recording. M5 for Delta weights: 1 (left),10 (right).

The second experiment would be the robustness in front to a change of microphone; which is scenario M2, and a simultaneous change of microphone and session scenarios M4, and M6. We computed the weighting parameters h_j on scenario M1, and tested on scenarios M2, M4 and M6. It can be seen in figure 10, that in the case of M2 where the recognition rates are already high, a near uniform weighting is better in the sense that the the use of a delta weight equal to one, gave a consistent improvement over all the phrases, while a high value of the delta weight, which is associated to a highly non uniform weighting lowered the recognition rate. On the other hand as can be seen in figure 11 and 12, where there is a the simultaneous change of session and microphone, the method proposed in the paper, yields a consistent improvement over a uniform weighting of each score and the globally best parametrization.

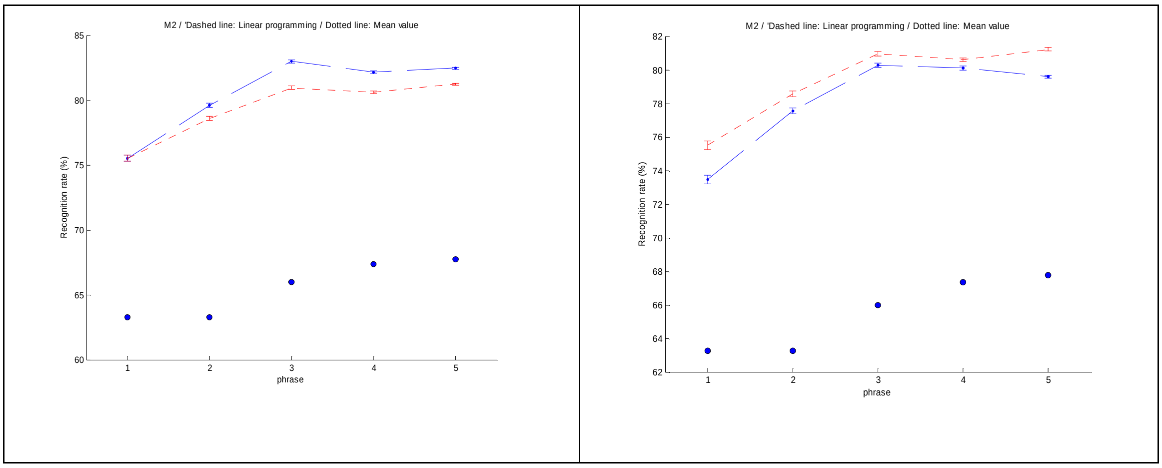

Figure 10: Robustness of the method with respect to a change in the microphone. Scenario M2 for Delta weights: 1 (left),10 (right).

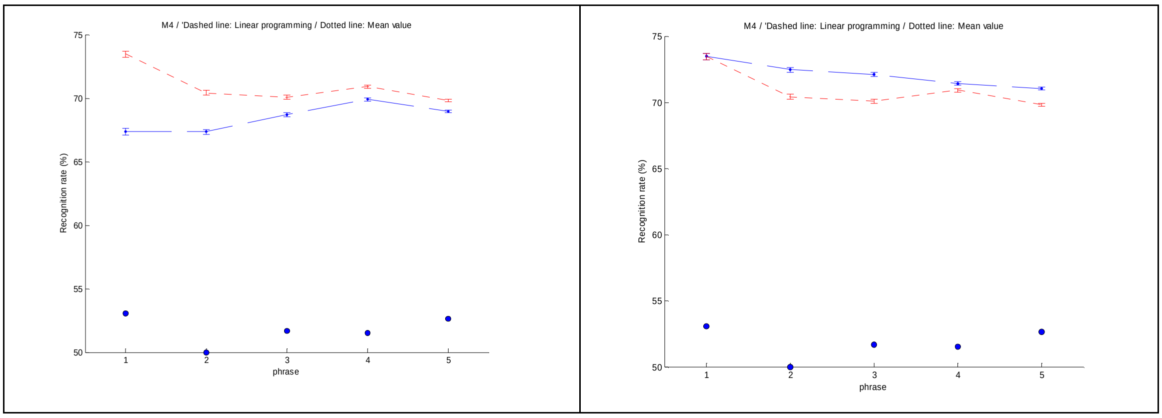

Figure 11: Robustness of the method with respect to a simultaneous change of microphone and session. Scenario M4 for Delta weights: 1 (left),10 (right).

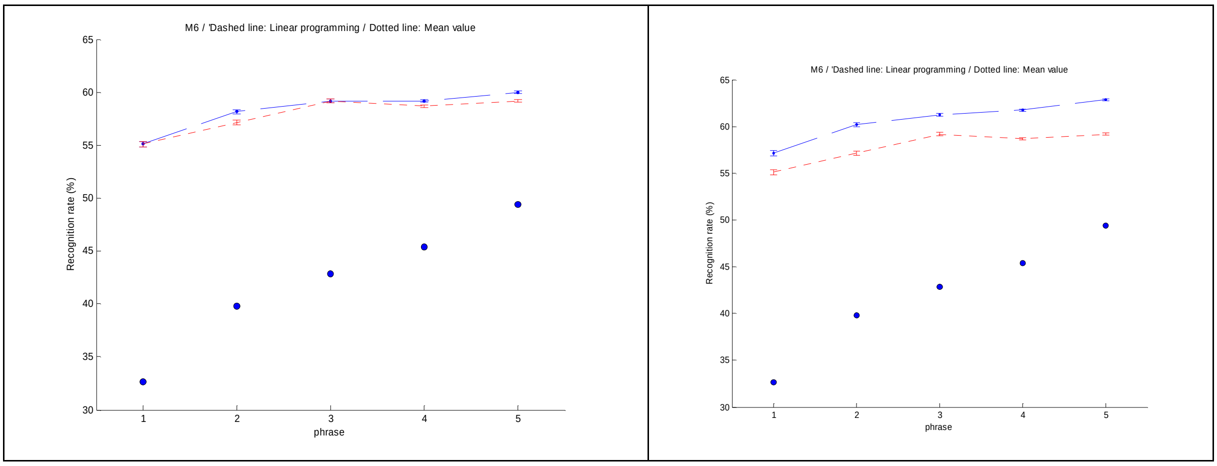

Figure 12: Robustness of the method with respect to a simultaneous change of microphone and session. Scenario M6 for Delta weights: 1 (left),10 (right).

4. CONCLUSIONS

We have presented a fusion method for likelihood model of the different channels to be fused. The method is based on a linear weighting of the log likelihood of the data given a model, and the weighting paramaters are estimated on a geometrical simplex. The algorithm for the maximum likelihood estimation of the weighting parameters was set as a linear programming problem, with a free parameters. The free parameter determines the uniformity of the weighting vector. The experiments showed that the presented fusion method gives robustness in front of a change of microphone and a change of session, i.e. the improvements were statistically significative with respet to a uniform weighting or to the best single parametrization.

ACKNOWLEDGEMENT

This work has been partially supported by FEDER and MEC, TEC2006-13141-C03-02/TCM, TIN2005-08852.

REFERENCES

Atal B.S. and Hanauer S.L., 1971. “Speech analysis and synthesis by linear prediction of speech wave," The Journal of the Acoustical Society of America, Vol. 50, 637–655.

Bellings, S.A., Fakhouri, S.Y., 1978. Identification of a class of nonlinear systems using correlation analysis. Proc. IEEE, 66 pp. 691-697.

Bertsimas, D. Tsitsiklis, J. N., 1997 *Introduction to Linear Optimization*, Athena Scientific (February 1, 1997)

Christopher M. Bishop, 1995, *Neural Networks for Pattern Recognition,* Oxford University Press

Boer, E.D., 1976. Cross-correlation function of a bandpass nonlinear network. Proc. IEEE, 64 pp. 1443-1444.

Boyd, S. and Vandenberghe, L. 2004 *Convex Optimization* Cambridge University Press.

Cover, T.M., Thomas J.A., 1991 Elements of Information Theory. Wiley Series in Telecommunications

Faundez M. and Rodriguez D., 1998. “Speaker recognition using residual signal of linear and nonlinear prediction models,” ICSLP, Vol. 2, 121–124.

Faundez-Zanuy M., Kubin G., Kleijn W.B., Maragos P., McLaughlin S., Esposito A.,. Hussain A, Schoentgen J., 2002. “Nonlinear Speech Processing: Overview and Applications,” Control and Intelligent Systems ACTA Press, 30,1, 1–10.

Faundez-Zanuy, M. 2004 “On the vulnerability of biometric security systems”. IEEE Aerospace and Electronic Systems Magazine. Vol.19 nº 6, pp.3-8, June.

Faundez-Zanuy, M. 2005a “Data fusion in biometrics” IEEE Aerospace and Electronic Systems Magazine. Vol.20 nº 1, pp.34-38, January.

Faundez-Zanuy, M. 2005b “Biometric recognition: why not massively adopted yet?” IEEE Aerospace and Electronic Systems Magazine. Vol.20 nº 8, pp.25-28, August.

Faundez-Zanuy, M., Monte-Moreno, E., 2005 “State-of-the-art in speaker recognition”. IEEE Aerospace and Electronic Systems Magazine. Vol.20 nº 5, pp 7-12, May.

Faundez-Zanuy, M. 2006 “Biometric security technology” IEEE Aerospace and Electronic Systems Magazine, Vol.21 nº 6, pp.15-26, June.

Hayakawa S., Takeda K. and Itakura F. 1997. “Speaker Identification Using Harmonic Structure of LP-Residual Spectrum," Audio Video Biometric Personal Authentification, LNCS 1206, 253–260.

He J, Liu L. and Palm G.. 1996. “On the Use of Residual Cepstrum in Speech Recognition," Proc. of IEEE ICASSP’96, Vol. 1, 5–8.

Jacoviti, G. , Neri, A., Cusani, R., 1987. Methods for estimating the autocorrelation function of complex stationary process. IEEE Trans. ASSP, 35, pp. 1126-1138

Kubin G., 1995. “Nonlinear processing of speech,” in Speech Coding and Synthesis (W.B. Kleijn and K.K. Paliwal), 557–610.

Ludmila I. Kuncheva 2004, *Combining Pattern Classifiers Methods and Algorithms*, John Wiley & Sons

Mahadeva Prasanna S. R., Cheedella S. Gupta and Yegnanaraya B., 2006. “Extraction of speaker-specific excitation from linear prediction residual of speech,” Speech Communication, 48, 1243–1261.

Mary L., Sri Rama Murty K., Mahadeva Prasanna S. R. and Yegna-Naraya B., 2004. “Features for Speaker and Language Identification,” Proc. of ISCA Tutorial and Research Workshop on Speaker and Language Recognition (Odyssey’04), 323–328.

Nikias, C.L. Petropulu, A.P., 1993. Higher-Order Spectra Analysis – A Nonlinear Signal processing Framework. Englewood Cliffs, NJ: Prentice-Hall.

Nikias, C.L., Raghuveer, M.R., 1987. Bispectrum estimation: A digital signal processing framework. Proc. IEEE, 75 pp. 869-890

Ortega-García, J., González-Rodríguez, J., Marrero-Aguiar, V., 2000. AHUMADA: a large speech corpus in Spanish for speaker characterization and identification. Speech Commun. 31 (June), 255–264

Prakriya, S. , Hatzinakos, D., 1985 “Blind identification of LTI-ZMNL-LTI nonlinear channel models”. Biol. Cybern., 55 pp. 135-144.

Reynolds, D.A. , Rose, R.C. , 1995 “Robust text-independent speaker identification using Gaussian mixture speaker models”, IEEE Trans. Speech Audio Process. 3 (1) 72–83.

Satue, A., Faundez-Zanuy, M. 1999 "On the relevance of language in speaker recognition" EUROSPEECH'99 Budapest, Vol. 3 pp.1231-1234

Solé-Casals, J. , Jutten, C. , Pham, D. T., 2005 “Fast Approximation of Nonlinearities”, Signal Processing vol. 85, pp. 1780-1786

Solé-Casals, J., Faudez-Zanuy, M., 2006 “Application of the mutual information minimization to speaker recognition/identification improvement”, Neurocomputing, vol.69, pp. 1467-1474

Solé-Casals, J., Jutten, C. , Taleb A., 2002 “Parametric approach to blind deconvolution of nonlinear channels”. Ed. Elsevier, Neurocomputing 48 pp.339-355

Taleb, A. , Jutten, C., 1999. Source separation in postnonlinear mixtures. IEEE Trans. on S.P., Vol. 47, n°10, pp.2807-20.

Taleb, A., Solé-Casals, J. , Jutten, C., 2001. Quasy-Nonparametric Blind Inversion of Wiener Systems. IEEE Trans. on S.P., Vol. 49, n°5, pp.917-924.

Thevenaz P. and Hügli H., 1995. “Usefulness of the LPC-Residue in Text-Indendent Speaker Verification,” Speech Communication, Vol. 17, no. 1-2, 145–157.

Yegnanaraya B., Reddy K.S., Kishore S.P., 2001. “Source and system features for speaker recognition using AANN models,” Proc. of IEEE ICASSP, 409–412 .

Zheng N. ; Lee T. ; Ching P.C., 2006. “Integration of Complementary Acoustic Features for Speaker Recognition,” IEEE Signal Processing Letters.