CENATAV
Centro de Aplicaciones de
Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2142
ISSN 2072-6287
Versión Digital

REPORTE TÉCNICO
Reconocimiento
de Patrones

SERIE AZUL

Topological Representation of Speech for Speaker Recognition

Ing. Gabriel Hernández Sierra,
Dr. C. José Ramón Calvo de Lara

RT_035 **octubre 2010**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
Ciudad de La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

CENATAV
Centro de Aplicaciones de
Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2142
ISSN 2072-6287
Versión Digital

REPORTE TÉCNICO
Reconocimiento
de Patrones

SERIE AZUL

Topological Representation of Speech for Speaker Recognition

Ing. Gabriel Hernández Sierra,
Dr. C. José Ramón Calvo de Lara

RT_035 octubre 2010

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
Ciudad de La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Topological Representation of Speech for Speaker Recognition

Ing. Gabriel Hernández Sierra, Dr. C. José Ramón Calvo de Lara

Centro de Aplicaciones de Tecnología de Avanzada, 7ª ·21812 e/ 218 y 222, Siboney, Playa, Habana, Cuba
ghernandez@cenatav.co.cu

RT_035 CENATAV
Fecha del camera ready: 30 de junio de 2010

Abstract: This research is in the field of speaker recognition and is based on the fact that behind each supervised or unsupervised learning algorithm, there is an implicit assumption about the structural nature of the data.

Our proposal is to find new speaker information using models of his speech. In order to characterize the speaker by his speech, a mapping function can be obtained from a set of speakers models, then projecting any new speaker in a new space which increases the variability between classes and decreases the variability intra classes, for its classification.

We will use topological space as a support to use inner geometric structure created by the projection of the acoustic models over a new spaces that reflects the nature of the speech acoustic classes of each speaker.

Keywords: Topological Representation, Speech, Speaker Recognition

Resumen: Esta investigación pertenece al campo de reconocimiento del locutor y se fundamenta en el hecho de que junto a cada algoritmo de aprendizaje supervisado o no supervisado hay una suposición sobre la naturaleza estructural de los datos.

Nuestro propósito es encontrar nueva información sobre el locutor usando modelos de su voz. Para caracterizar al locutor por su voz una función de mapeo puede obtenerse a partir de un conjunto de modelos de locutores, entonces proyectando cualquier nuevo locutor en un nuevo espacio se incrementa la variabilidad entre clases y disminuye la variabilidad intraclases, para su clasificación.

Usamos espacios topológicos como base para usar la propia estructura geométrica creada por la proyección de los modelos acústicos sobre un nuevo espacio que refleja la naturaleza de las clases acústicas de cada locutor.

Palabras clave: Representación Topológica, Voz, Reconocimiento del Locutor

1 Introduction

During the last decade researchers in the field of text independent speaker recognition have developed statistical classifiers, discriminatory classifiers and mixtures of them, all working over the same model, without using the topological information existing in data, which encourages us to search for new information that would be able to improve the recognition results until today. An example of this information could be obtained by a new mapping function that contains topological information of the Gausian Mixture Models “GMM”, to project these models in a new space where the mixtures of the same speaker are in a reduced neighborhood and models of other speakers are in different neighborhood.

The main goal of our work is focused specifically on the task of model training and classification – two key stages in a speaker recognition system- with increased robustness to channel and session mismatch.

2 State of the Art of model training in Speaker Recognition

Training

In GMM speaker model [1], each speaker is represented by the means, covariance, and weights of a mixture of C multivariate diagonal-covariance Gaussian densities defined in some continuous speech features space of dimension F. Baseline algorithm for speaker model training is GMM-MAP algorithm [2], which is a maximum a posterior “MAP” adaptation of the classical GMM [3] adapting the Gaussian density components from a Universal Background Model “UBM” [4]. Classical MAP adaptation is by far the most popular type of speaker modeling in text-independent speaker recognition. A state-of-the-art GMM–UBM system was submitted to NIST’04 SRE [6] by LIA (University of Avignon) and is described in [5].

Two recent compensation methods have a crucial impact on overall performances, Nuisance attribute projection “NAP” [7] and factor analysis “FA” [8].

Joint Factor Analysis “JFA” [9], is a method to include speaker and session variability factors in GMM, has become the state of the art in the field of speaker verification during last years, this analysis plays a key role to estimate a compensate target speaker model.

The basic assumption in JFA is: let C be the number of components in a GMM-UBM and F the dimension of the acoustic feature vectors. The CF-dimension vector M obtained by concatenating the F-dimensional GMM’s mean vectors corresponding to a given utterance is called supervector.

The assumptions in this algorithm are as follow. First, we assume that a speaker and channel-dependent supervector M can be decomposed into a sum of two supervectors, a speaker supervector s, and channel supervector c

$$M=s + c \tag{1}$$

where s and c are statistically independent and normally distributed.

Second, we assume that the distribution of s has a hidden variable description of the form

$$\mathrm{s}=m + vy + dz \tag{2}$$

where m is a CF supervector, v is a rectangular matrix of low rank and y is a normally distributed random vector, d is a CFxCF diagonal matrix, and z is a normally distributed CF-dimensional random vector. We will refer to the columns of v as eigenvoices.

Third, we assume that the distribution of c has hidden variable description of the form

$$c=ux \tag{3}$$

where u is a rectangular matrix of low rank, and x is a normally distributed random vector. We refer to the components of x as channel factors, and we call the columns of u as eigenchannels.

The assumption in eq. (2) is equivalent to say that s is normally distributed with mean m and covariance matrix $d^2 + vv^*$, eq. (2) is a model of inter speaker variability. If v=0 and u=0, the assumption in eq. (2) is the same as in classical MAP [10]; on the other hand, if d=0 and u=0, the assumption is the same as in eigenvoices MAP [11].

3 Classification

For text-independent speaker verification, the GMM based on statistical theory is the most widely used method. A UBM is usually trained from a very large set of development data comprised of various speakers, channel types and number of sessions. Target speaker models can be obtained by adapting the UBM according to their corresponding enrollment data. A GMM-UBM system computes a likelihood ratio score for an unknown test utterance between a speaker-dependent acoustic distribution (GMM-MAP model) and a speaker-independent acoustic distribution (UBM model).

Support vector systems “SVM” were first proposed in the early 1990s as optimal margin classifiers in the context of Vapnik’s statistical learning theory [12]. Since then have become an important alternative of GMM and is widely used as a discriminative classification technique in the field of speaker recognition. With the proper use of appropriate sequence kernels, such as generalized linear discriminant sequence “GLDS” kernels [13] or Fisher Kernels [14], SVM systems can obtain comparable performances to GMM-UBM systems with relatively moderate computation complexity. Recently, a new SVM-based speaker verification strategy using GMM-UBM super vectors has been proposed by Campbell [13] to combine the advantages of the two systems. These CF-dimensional super vectors are obtained by stacked the means of the GMM-UBM mixture components (See fig. 1).

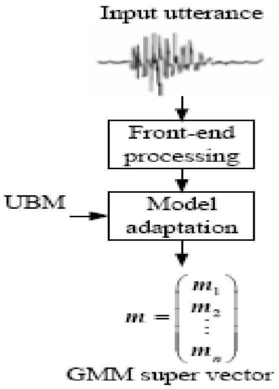


Fig. 1. Procedure of GMM super vector generation

A GMM-SVM system is a combination of a GMM-UBM and a SVM system. The GMM-UBM system serves as a feature extractor for the attached SVM system. The SVM classifier is used to model the target speaker characteristics and to score the test utterances (See fig. 2).

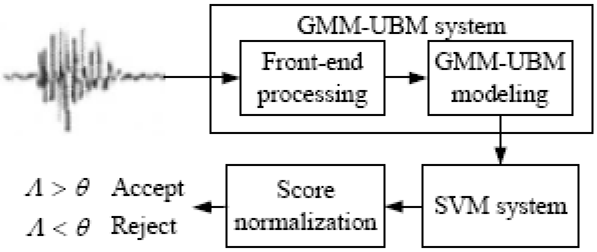


Fig. 2. Structure of a GMM-SVM system

Linear classification techniques are then applied in this potentially high-dimensional space. The main component in an SVM is the kernel, which is an inner product in the SVM feature space. Since inner products induce metrics distance, the basic goal of SVM kernel design, is to find an appropriate metric in the SVM feature space relevant to the classification problem.

SVM is a two-class classifier. In the standard formulation, an SVM, f (x), is given by eq. 4.

$$f(x) = \sum_{i=1}^{L} \alpha_i t_i K(x, x_i) + d \tag{4}$$

where 〈 · ,· 〉 stands for the inner product and K(x ,xi) is a kernel function; the vectors x are the training set, the vectors xi are support vectors obtained from the training set; ti are the ideal outputs, either 1 or -1, depending upon whether the corresponding support vector is in class 0 or class 1 (positive or negative), respectively. $\sum_{i=1}^{L} \alpha_i t_i = 0, \ \alpha_i . > 0$.

The kernel K(., .) is constrained to have certain properties (the Mercer condition), so that K(.,.) can be expressed as:

$$K(x, x_i) = \phi(x)^t \phi(x_i) \tag{5}$$

where Φ(x) is a mapping from the input space (where x lives) to a possibly infinite-dimensional SVM expansion space. The data points from the training set lying on the boundaries are the support vectors xi in equation (4). The focus of the SVM training process is to model the boundary between classes. For a separable data set, SVM optimization chooses a hyperplane in the expansion space with maximum margin [12].

SVM is trained for each target speaker using the GMM super vector of the speaker's enrollment utterances as positive samples, and GMM super vectors of all utterances from background speakers as negative samples.

For classification, a class decision is based upon whether the value, f(x), is above or below a threshold.

4 News approaches about topological voice representation

The following articles reflect new approaches of topological voice representation in the field of speech recognition:

Authors: Andrew Errity and John McKenna

- Institution: School of Computing, Dublin City University, Ireland

1. An Investigation of Manifold Learning for Speech Analysis (2006) [15].

This paper assumes that the acoustic sounds lie on a high-dimension manifold, so if the range of the acoustic sounds of the human voice is a subset of all sounds, therefore it lies on a low-dimensional submanifold of the high-dimension space of all possible sounds. This work carried out a study of a number of algorithms for learning about the manifold of voice in an effort to extract the inner geometric structure of the voice signal. For this, they evaluated the ability of manifold learning algorithms to separate vowels in a low-dimensional space compared to a classical linear dimensionality reduction method.

The manifold algorithms used were:

- Locally linear embedding “LLE” [16, 17]
- Isometric feature mapping “Isomap” [18]
- Laplacian Eigenmaps [19]

These algorithms will be explained in the next section.

The figure 3 was drawn from article.

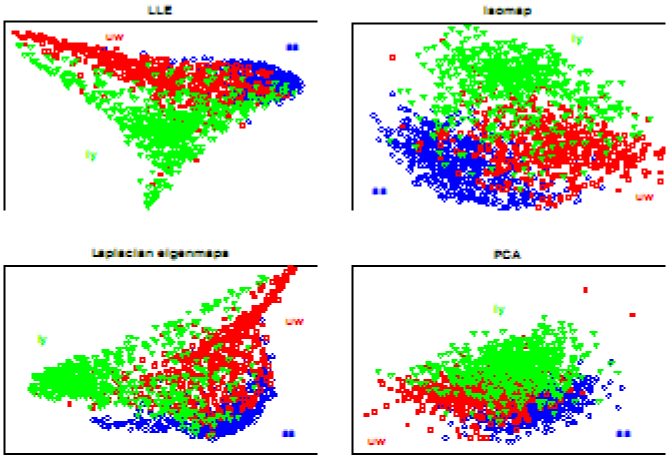


Fig. 3. 600 tokens of each the vowels ‘*aa*’, ‘*uw*’ and ‘iy’ within the two dimensional vowel space produced by *LLE, Isomap, Laplacian eigenmaps* and *PCA*

These results indicate that manifold algorithms work better than traditional linear methods, separating vowels in a low-dimensional space and are able to discover a useful geometric structure of the voice.

Authors: Aren Jansen and Partha Niyogi

- Institution: The University of Chicago Dept. of Computer Science and Statistics, USA

1. Intrinsic Fourier Analysis on the manifold of speech sounds (2006). [20]
2. A Geometric Perspective on Speech Sounds (2005). [21]
3. The Manifold Nature of Vowel Sounds (2007). [22]

These three papers have a main goal in common, to motivate and demonstrate the existence of a low-dimensional curved manifold structure in voiced speech sounds, and each one have a second goal, independently.

- The first introduces a new spectrogram technique based on the existence of this manifolds structure. Also concludes that this representation allows a better phonetic distinction in a low-dimensional space.
- The second exploits the implications of this geometric point of view in human speech and shows that manifold structure of speech sounds may be exploited for dimensionality reduction, semi-supervised learning and speech representation.
- The third presents a comparison of representation of phonemes / a / and / ae / using a manifold-based Laplacian eigenmap dimensionality reduction algorithm and a traditional Principal Component Analysis.

In these articles we focus on primary objective: the study of the existence of inner geometric structure in the speech signal.

The existence of a manifold structure of the speech signal, as shown in previous articles, has several implications and new related applications, for example:

- If the voice sounds lie in a manifold, representation and classification of speech require the estimation of functional maps whose domain is this manifold. This motivates the use of Laplacian Eigenmaps [19] to obtain a suitable basis of these intrinsically defined functions. This may be used for dimensionality reduction of speech features and to define alternative spectrograms that exploit this underlying geometric structure.
- The manifold structure of the speech sound does not reflect a linear relationship between the articulatory space and the acoustic space. Therefore, alternative metric distances (such as the geodesic distance), must be considered.

5 Manifold definitions

Before continuing, some needed definitions must be done:

Definition 1. A topological space X consists of a set X together with a collection of subsets ϕ, referred to as open sets, such that the following conditions are satisfied:

(i) the empty set ϕ and the whole set X are open sets,

(ii) the union of any collection of open sets is itself an open set,

(iii) the intersection of any *finite* collection of open sets is itself an open set.

The collection of all the open sets in a topological space X is referred to as a topology on the set X and denotes by (X,τ).

Definition 2. A base of a topology is a family of open subsets such that any other open set may be represented as the union of subsets constituting the base of the topology.

Definition 3. Let (X,τ) and (Y,S) be topological spaces; a mapping $f: X \rightarrow Y$ is called continuous if and only if $f^{-1}(U) \in \tau$ for each $U \in S$ the inverse image of any open subset of Y is open in X.

Definition 4. A homeomorphism between two topological spaces M and N is a bijective (=one-to-one) map f : M → N such that both f and f -1 are continuous (with respect to the topologies of M and N).

Definition 5. A topology is Hausdorff, if for any two distinct points x1; x2 there exist two disjoint open subsets U_j, and $x_j \in U_j, j = 1,2.$

Definition 6. Let M be a topological Hausdorff space with a countable basis. M is called a topological manifold if there exists an $m \in \aleph$ and for every point $p \in M$ an open neighbourhood Up of p, such that Up is homeomorphic to some open subset Vp of $\Re^m$. The natural number m is called the dimension of M. This means that a topological manifold Mm is locally homeomorphic to the standard m-dimensional vector space $\Re^m$.

A manifold is a space that is locally like $\Re^m$, however lacking a preferred system of coordinates. Furthermore, a manifold can have global topological properties.

6 Manifold learning algorithms

Jansen and Niyogi [21] shown recently that acoustic features lie in a low-dimension manifold that is embedded in an acoustic space of high-dimension. A low-dimension submanifold can have a highly non-linear1 structure that linear methods would not be able to discover. A series of manifold learning algorithms (also known as non-linear algorithms, to reduce the dimensionality) have been proposed [16, 18, 19], which go beyond the limitations of linear methods.

The steps followed by the manifolds learning algorithms mentioned above (LLE, Isomap, and Laplacian Eigenmaps) begin creating a graph and then reducing the dimensionality.

Assume we have a dataset in a N × D matrix X consisting of N datavectors xi, (i = 1, 2, . . . , N) with dimensionality D. Assume further that this dataset has intrinsic dimensionality d (where d < D, and often d << D). Here, in mathematical terms, intrinsic dimensionality means that the points in dataset X are lying on or near a manifold with dimensionality d that is embedded in the D-dimensional space.

6.1 Isometric feature mapping: "Isomap"

The key assumption made by Isomap [18] is that the distance along the curve between two points is not the straight that connect them, but the shortest path through the points on the curve that connect them. The basic idea is to construct a graph whose nodes are the data points, where a pair of nodes is adjacent only if the two points are close in RD(D is the dimension of the data), and then take the geodesic distance along the manifold between any two points as the shortest path in the graph and finally to use multidimensional scaling "MDS" -a classical method for embedding dissimilarity information into Euclidean space [24]- to extract the low dimensional representation (as vectors in $R^d, d << D$).

[1] If we talk about complex spatial relationships, we talk about that the relationship between the data are non-linear, and then we can say that the data lie on a non-linear manifold.

6.2 Locally linear embedding: "LLE"

LLE was introduced at about the same time as Isomap. Unlike previous methods, LLE recovers global nonlinear structure from locally linear fits. This algorithm is based on simple geometric intuitions. Suppose the data consist of N real-valued vectors $\hat{X}_i$, ($i = 1, 2, \cdots, N$), each of dimensionality D, sampled from some underlying manifold. LLE starts with finding the k nearest neighbors (based on the Euclidean distance) for each vector $\hat{X}_i$, $1 \le i \le N$. Let Ni denote the indices of the k nearest neighbours of the vector $\hat{X}_i$.

LLE finds the optimal local convex combinations of the k-nearest neighbors to represent each original vector. It is equivalent to minimizing the sum of the reconstruction errors ei.

$$\varepsilon(W) = \sum_i e_i \tag{6}$$

where $e_i \equiv \left| \hat{X}_i - \sum_{j \in N_i} W_{ij} \hat{X}_j \right|^2$ should be unaffected by any global translation $\hat{X}_i \to \hat{X}_i + \delta$, $\delta \in R^D$, give the condition $\sum_{j \in N_i} W_{ij} = 1 \ \forall i$, where W is a edge weights matrix. Note that each e_i is also invariant to global rotations and reflections of the coordinates.

Then, LLE considers a projection space that has a dimension much smaller than D. Let $\hat{Y}_i$ be the projection of $\hat{X}_i$ in the projection space. The projections $\hat{Y}_i$ □are chosen such that the following objective function is minimized:

$$\phi(Y) = \sum_i \left| \hat{Y}_i - \sum_{j \in N_i} W_{ij} \hat{Y}_j \right|^2 \tag{7}$$

Note that the above is equivalent to finding a lower dimensional representation, such that the local convex representations are preserved. It can be shown that with some additional conditions, which make the problem well defined, the minimization task can be accomplished by solving a sparse N x N eigenvector problem [25]. More specifically, the d eigenvectors associated with the d smallest non-zero eigenvalues provide an ordered set of orthogonal coordinates centered on the origin.

LLE Algorithm

1. Compute the *k* nearest neighbors of each point $\hat{X}_i$
2. Compute the weights W_{ij} of a convex combination of the *k* nearest neighbors that best represent the point $\hat{X}_i$.

3. Find a low-dimensional projection $\hat{Y}_i$ such that the above local representations are best preserved.

6.3 Laplacian Eigenmaps

Laplacian Eigenmap algorithm, proposed by Belkin and Niyogi [19], is based on ideas from spectral graph theory. This work establishes both a unified approach to dimension reduction and a new connection to spectral theory.

We describe the Laplacian Eigenmap for discrete data. Again, we consider N vectors, $\hat{X}_i$, ($i = 1,2,\cdots,N$), in the D-dimensional data space . For each vector $\hat{X}_i$, $1 \le i \le N$, suppose a neighbour vector set Ni is computed. A graph identical with the graph in ISOMAP can be defined. For any pair of connected points $\hat{X}_i$ and $\hat{X}_j$, we define a "local similarity" matrix W which reflects the degree to which points are near to one another. There are two choices for W:

1. $W_{ij} = 1$, if $\hat{X}_j$ is one of the k-nearest neighbours of $\hat{X}_i$; 0 otherwise (simple weight scheme).
2. $W_{ij} = e^{-\|\hat{X}_i - \hat{X}_j\|^2 / 2\sigma^2}$, for neighbouring nodes; 0 otherwise. This is the Gaussian heat kernel, which has an interesting theoretical justification given in [26]. On the other hand, use of heat kernel requires manually setting σ, so is considerably less convenient than the simple weight scheme.

Let D denote a diagonal matrix such that $D_{ii} = \sum_j W_{ji}$. Let W denotes the symmetric matrix with entries, $1 \le i, j \le N$. Finally, given a graph and a matrix of edge weights, W, the Laplacian graph is defined as $L \overset{def}{=} D - W$. Consider the solutions to the problem:

$$Lf = \lambda Df \tag{8}$$

where $f \in \Re^N$. Let $f_0, f_1, \cdots, f_{k-1}$ be the eigenvectors with corresponding eigenvalues $0 = \lambda_0 \le \lambda_1 \le \lambda_2 \le \cdots \lambda_{k-1}$;

$$Lf_0 = \lambda_0 Df_0,$$

$$Lf_1 = \lambda_1 Df_1,$$

$$\vdots$$

$$Lf_{k-1} = \lambda_{k-1} Df_{k-1}.$$

The eigenvectors associated with zeros eigenvalues is left out and the next m eigenvectors are used for the embedding in an m-dimensional Euclidean space.

The eigenvalues and eigenvectors of the Laplacian reveal a wealth of information about the graph such as whether it is complete or connected. Here, the Laplacian will be exploited to capture local information about the manifold.

Laplacian Eigenmaps Algorithm

1. Set $\left\{ \begin{array}{l} W_{ij} = \left\{ \begin{array}{cc} e^{-\left\|\hat{X}_i-\hat{X}_j\right\|^2 / 2\sigma^2} & if\ \hat{X}_j \in N_i \\ 0 & otherwise \end{array} \right\} \\ or \quad W_{ij} = \left\{ \begin{array}{ll} 1 & if\ \hat{X}_j \in N \\ 0 & otherwise \end{array} \right\} \end{array} \right\}$
2. Let F be the matrix whose columns are the eigenvectors of $Lf = \lambda Df$ with nonzero accompanying eigenvalues.
3. Return $Y := [F]_{m \times N}$.

Local/Global Methods

Manifold learning algorithms are commonly split into two camps: local methods and global methods. Isomap is considered a global method because it constructs an embedding from the geodesic distance between all pairs of points. LLE is considered a local method because the cost function that it uses to construct an embedding only considers the placement of each point with respect to its neighbours. Similarly, Laplacian Eigenmaps and the derivatives of LLE are local methods.

The local/global split reveals some distinguishing characteristics between Manifold learning algorithms: local methods tend to characterize the local geometry of manifolds accurately, but break down at the global level. For instance, the points in the embedding found by local methods tend to be well placed with respect to their neighbours, but non-neighbours may be much nearer to one another than they are on the manifold.

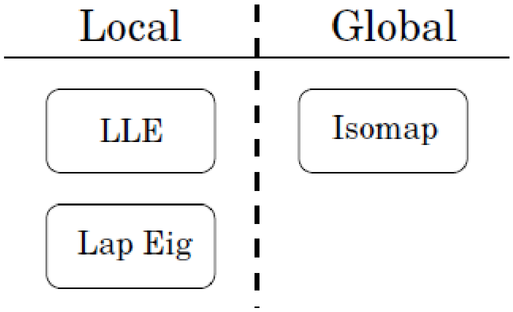


Fig. 4. The local/global split

The behaviour for Isomap is just the opposite: the intra-neighbourhood distances for points in the embedding tend to be inaccurate whereas the large interpoint distances tend to be correct.

The local methods also seem to handle sharp curvature in manifolds better than Isomap. Additionally, shortcut edges – edges between points that should not actually be classified as neighbours – can have a potentially disastrous effect in Isomap. A few shortcut edges can badly damage all of geodesic distances approximations. In contrast, this type of error does not seem to propagate through the entire embedding with the local methods.

7 Manifolds in Speaker Recognition

Motivations

The idea of using the internal geometric structure of the data to improve the data processing algorithms is increasingly used in Machine Learning. In fact, most of the problems of learning in real life show a geometric structure. An astute use of geometric structure, through a nonlinear dimensionality data reduction - recovering meaningful low-dimensional structures hidden in high-dimensional data -, should improve the performance of any learning algorithm.

To provide a motivation for using a manifold structure, consider a simple example shown in figure 5 [16]. The two classes consist of two parts of the curve shown in the first panel (row 1). We are given a few labeled points and a 500 unlabeled points shown in panels 2 and 3 respectively. The goal is to establish the identity of the point labeled with a question mark "?". There are several observations that may be made in the context of this example:

1. By observing the picture in panel 2 (row 1) we see that we cannot confidently classify by using the labeled examples alone. On the other hand, the problem seems much more feasible given the unlabeled data shown in panel 3.
2. Since there is an underlying manifold, it seems clear at the outset that the geodesic distances along the curve are more meaningful than Euclidean distances in the plane. Many points which happen to be close in the plane are on the opposite sides of the curve. Therefore rather than building classifiers defined on the plane it seems preferable to have classifiers defined on the curve itself.
3. Even though the data suggests an underlying manifold, the problem is still not quite trivial since the two different parts of the curve come confusingly close to each other. There are many possible potential representations of the manifold and the one provided by the curve itself is unsatisfactory. Ideally, we would like to have a representation of the data which captures the fact that it is a closed curve. More specifically, we would like an embedding of the curve where the coordinates vary as slowly as possible when one traverses the curve. Such an ideal representation is shown in the panel 4 (first panel of the second row). Note that both represent the same underlying manifold structure but with different coordinate functions. It turns out (panel 6) that by taking a two-dimensional representation of the data with Laplacian Eigenmaps [23], we get very close to the desired embedding. Panel 5 shows the locations of labeled points in the new representation space.We see that "?" now falls squarely in the middle of "+" signs and can easily be identified as a "+".

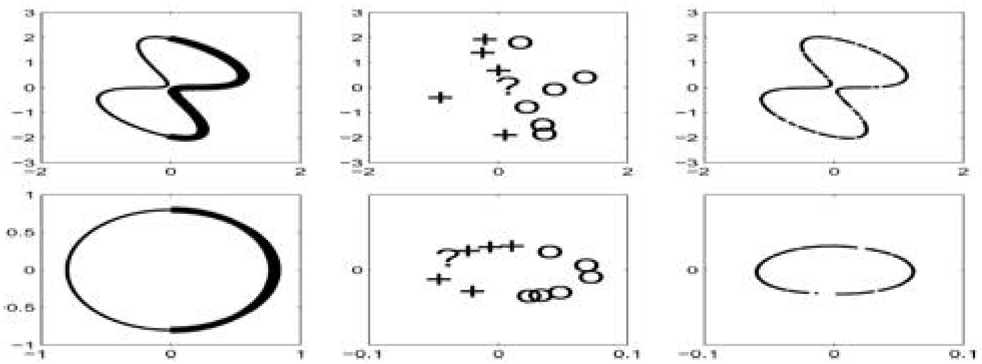


Fig. 5. Top row: Panel 1. Two classes on a plane curve. Panel 2. Labeled examples. "?" is a point to be classified. Panel 3. 500 random unlabeled examples. Bottom row: Panel 4. Ideal representation of the curve. Panel 5. Positions of labeled points and "?" after applying eigenfunctions of the Laplacian. Panel 6. Positions of all examples

To study the data nature, we need tools capable of projecting the data in a space where we can study its geometric structure. For this, we need the projection to keep the relations between the neighbours, and the relations between the neighborhoods of the general structure. Therefore the algorithms described above were used in this work.

This geometric point of view leads to new and exciting ideas in the areas of speech recognition, speech signal representation and speaker recognition.

7.1 Speaker topologic information

From the previous study, we are motivated to use the manifolds or Hausdorff topological spaces in our research, by improving the performance in the area of speaker recognition.

Our work will be aimed at the following:

- To obtain a new speaker information of the speech named speaker topology information.
- To evaluate this new representation in text independent speaker identification or verification.

The main reason for using topology information in speaker recognition is:

- The intuitive idea that the individual components of a multi-modal density (GMM) are capable of modeling the underlying acoustic classes of the acoustic space that characterizes the voice of an individual, which can be approximated by a set of acoustic classes (representing large groups of acoustic events) such as the vowels, consonants or nasals, fricatives, etc [23]. These acoustic classes show dependence on the configurations of the vocal tract of each speaker, so they are very useful in characterizing a speaker. Clustering is a grouping strategy in the maximum-expectancy (EM) algorithm used to obtain the GMM-UBM adapted. Although the clusters models do not work well in applications where the data shows a kind of very complex spatial relationship, the clusters models are used in this GMM space. Then, an intelligent use of geometric structure that is found in the speech should improve the performance of any speaker recognition algorithm.
- The Joint Factor Analysis (JFA) [9] approach has become during the last years the state of the art in the field of speaker verification. This algorithm is used to attack the

> problem of speaker and channel variability in GMM framework. The combination of JFA and SVM is used in speaker verification [13], in this approach the SVMs uses as input data the super vectors obtained by simple stacked means of the JFA model. These CF-dimensional super vectors are obtained by simple stacked of C points, each one F-dimensional, which are the means of the acoustic classes, these means lie on a low-dimensional topological manifold structure that does not reflect a linear relationship as shown in [19-22].

It is also important to note that while objects (points) are typically represented by vectors in R^n, the real distance is often different from the distance induced by the ambient space R^n (usually the Euclidean distance).

Hypothesis

Gaussian components are typically embedded in a very high dimensional space. However, the intuition of researchers always has been that although these components are ostensibly high dimensional, there is a low dimensional structure inner it that needs to be discovered. If we utilize the inner geometric structure of the underlying space of Gaussian components to construct a low dimensional representations, where the components distribution are kept in a good condition, and in addition contain new topologic information, should improve the performance of speaker recognition algorithm.

Note that, if this low dimensional representation is a linear subspace, then linear projections to reduce the dimensionality are sufficient. Classical techniques like Principal Components Analysis and Random Projections may be used to construct classifiers in the new low dimensional space. On the other hand, if the components lie on a low dimensional manifold embedded in the higher dimensional space, then is necessary do something different.

7.2 Initial speech data for representation

Gaussian components lie in a manifold, based on this assumption the next step would be to build a graph that captures the geometric structure of the acoustic features of speech signal. To construct the graph that represents the nature of the speaker's voice we use the information obtained from the GMM-UBM adaptation of the acoustic features, specifically the mean matrix.

Gaussian Mixture Model (GMM)

$$\lambda = \{p, \mu, \Sigma\}$$

$\bar{p} \rightarrow$ weight vector

$\mu \rightarrow$ mean matrix

$\sum \rightarrow$ covariance matrix.

The mean matrix X are drawn from an speech expression, and can be represented as a F x C matrix, were F is the dimension of acoustic feature and C is the number of Gaussian components.

$$X = \left\{ \begin{array}{cccc} \underline{\vec{x}_1} & \underline{\vec{x}_2} & \cdots & \underline{\vec{x}_C} \\ \downarrow & \downarrow & & \downarrow \\ x_{1,1} & x_{1,2} & \cdots & x_{1,C} \\ x_{2,1} & \cdots & & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_{F,1} & \cdots & & x_{F,C} \end{array} \right\} \text{Mean matrix}$$

7.3 Algorithm generalization

To generalize our approach we describe the algorithm using the Laplacian Eigenmaps.
First, given $X_{F,N}$ initial mean matrix, where *N* is the number of observation, each *n*, *(n=1,2,…,N)*, is considered one point in *F*-dimensional data space. For each point $\hat{x}_n$, $1 \le n \le N$, the k-nearest neighbours V_n is computed using the Euclidean distance. Let V_n and V_z denote the index sets of the points that are the neighbor of $\hat{x}_n$, and $\hat{x}_z$.Then, we construct a graph where each $\hat{x}_n$, is a vertex and two vertices has an edge if and only if $\hat{x}_n \in V_z$ or $\hat{x}_z \in V_n$. For any pair of connected points $\hat{x}_n$, and $\hat{x}_z$ be define weight function $W_{n,z} = 1$. Let A denote a diagonal matrix such that $A_{n,n} = \sum_z W_{n,z}$, *W* denote a symmetric matrix with entries $W_{n,z}$.

Then, the Laplacian matrix is computed $L = A - W$, finally we compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$Lf = \alpha A f \tag{9}$$

where $f \in \Re^N$, Let $f_0, f_1, \cdots, f_{k-1}$ be the solution vectors with corresponding eigenvalues $0 = \alpha_0 \le \alpha_1 \cdots \le \alpha_{k-1}$.

The eigenvectors associated with zeros eigenvalues is left out and the next k eigenvectors are used for the embedding in a k-dimensional Euclidean space.

A map from the manifold to $\Re$ is obtained as a result of applying Laplacian Eigenmaps method, let f be a map from the manifold to $\Re$, $f : X^F \to \Re^k$.

Given that f is a eigenvectors set we can define Mk,N as the matrix whose columns are these k eigenvectors, $k << F$.

After obtaining the mapping function f of the initial matrix XF,N we need to obtain a base change matrix B between XF,N and Mk,N. For this we have to resolve the following equations system:

$$X_{F,N} * B \approx M_{k,N} \tag{10}$$

Note that this process can be performed for any of the above described manifold learning algorithms.

7.4 Nature of the supervector for speaker recognition

The supervectors are built from the concatenation of the center of the Gaussian components for each speaker model; the centers of Gaussian components in each model are represented as a means matrix Mi, which is concatenated for columns in a supervector, as follows:

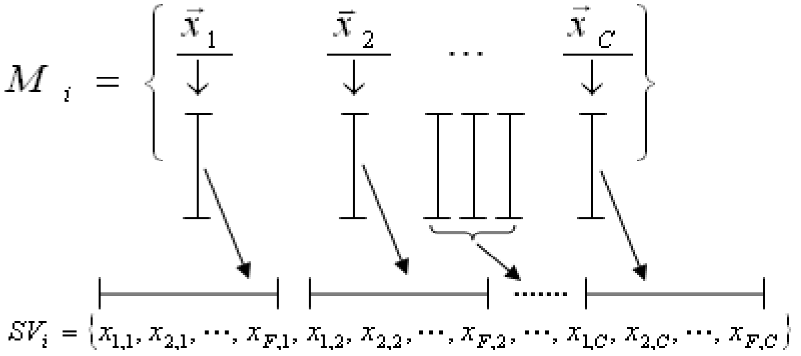
$$M_i = \left\{ \begin{array}{cccc} \underline{\vec{x}_1} & \underline{\vec{x}_2} & \cdots & \underline{\vec{x}_C} \\ \downarrow & \downarrow & & \downarrow \\ x_{1,1} & x_{1,2} & \cdots & x_{1,C} \\ x_{2,1} & \cdots & & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_{F,1} & \cdots & & x_{F,C} \end{array} \right\} \text{ Mean matrix}$$

$$SV_i = \{x_{1,1}, x_{2,1}, \cdots, x_{F,1}, x_{1,2}, x_{2,2}, \cdots, x_{F,2}, \cdots, x_{1,C}, x_{2,C}, \cdots, x_{F,C}\} \text{ mean super vector “SV”}$$

where *i* represents the index of each speaker.

This process is only a transformation of the model mean matrix to a supervector SV with FxC-dimensions, converting the matrix into a point on a high-dimensional space which will be used later to classify the speaker through a SVM classifier.

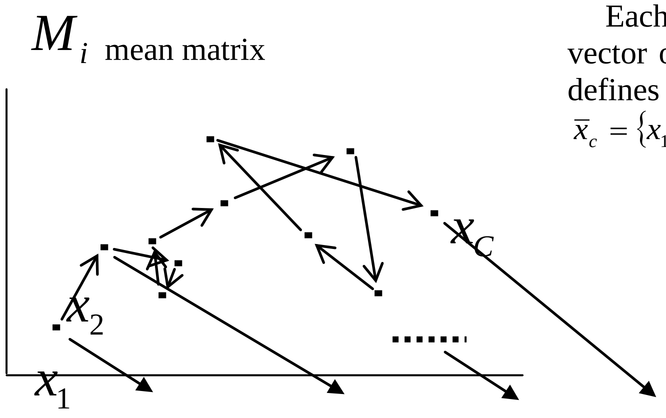
The following scheme graphically illustrates the process.


$M_i =$
$\vec{x}_1$
$\vec{x}_2$
$\cdots$
$\vec{x}_C$
....
$SV_i = \{x_{1,1}, x_{2,1}, \cdots, x_{F,1}, x_{1,2}, x_{2,2}, \cdots, x_{F,2}, \cdots, x_{1,C}, x_{2,C}, \cdots, x_{F,C}\}$

where *i* represents the index of each speaker.

Note that as a result of this construction each Gaussian components defines a specific set of dimensions in the SV and the union of all components defines the place of the speaker in a high-dimensions space.

To better address future work we will illustrate from another point of view the construction of the SV taking into account that each column c (1,2,…C) in the Mi matrix correspond to each Gaussian component:



Each column represents a means vector of a Gaussian component which defines an acoustic class. $\overline{x}_c = \{x_1, x_2, \cdots, x_F\}$,

where *i* represents the index of each speaker.

Note that the order in which means vectors of the Gaussian component are chosen in the construction of the SV is not important if the same order is used always for all speakers.

Then we show in fig 6 as seen a group of means of the Gaussian components of several speakers in only two dimensions, each color representing a different acoustic class, each dot in an acoustic class is a different speaker.

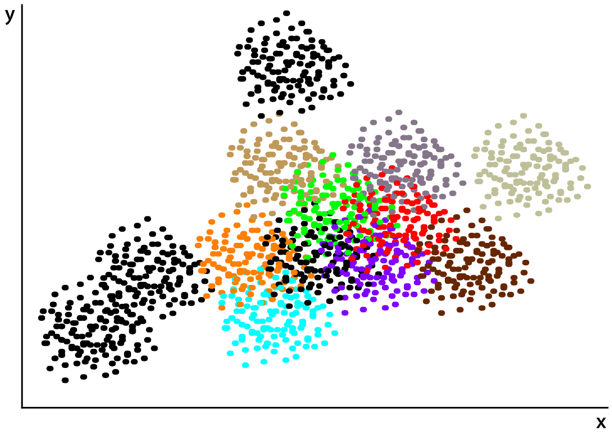


Fig. 6. Group of means of the Gaussian components of several speakers in two dimensions (x,y)

From this representation, combined with previous studies a group of potential ways to improve the performance of the speaker classifiers are proposed:

a) A local analysis for each subspace or submanifold of acoustic components looking for a new representation which increases the distance between the means of the same acoustics class for different speakers, and decreases the distance between the means of the same speaker, will allow each submanifold corresponding to this acoustics class in the SV to be more discriminative.

b) A global analysis of the manifold where all the acoustic classes lie, to find a new representation in a linear space where the geometric information of the topological structure of these acoustic classes is taken into account, will allow greater discriminative power for each SV, built from the new space.

c) All researchers in the field known that there is a lot of redundant information in the speaker acoustic models obtained using the adapted GMM-UBM, which can be observed easily increasing the amount of mixtures in the Universal Background Model and as result we will obtain an improvement in the EER. This is adversely not proportional to the big size reached by the SV of each speaker, which often generates more problems than the small benefits in percentage of EER improvement achieved. An analysis of the figure 6, which represents a similar nature to the space where the acoustic classes exist, shows that in the center of the cloud of points there is greater overlap between acoustic classes and if the number of classes increases, the distance between them respect to the center of this cloud decreases delicately, in addition to this analysis we also rely on our earlier work see [30]. So if we use the topologic information of acoustic classes we can reduce the dimension of these classes and the supervector size, without greatly affecting the outcome of the classifier.

From the above analysis we developed three experiments which are described below.

8 Speaker verification experiment

Three experiment using speakers of NIST’04 for background data (UBM), 1348 speakers of NIST’05 for development data and 380 speakers of Fisher’04 for impostor data were done, in order to evaluate the proposed algorithms. The experimental protocol is fixed throughout the work. The male section of the NIST’05 primary task is used for development. For this condition, one side of a 5-min conversation is given for testing and the same amount for trainng.

The realization has taken place in the context of the open-source ALIZE toolkit [31, 32] and the algorithms Isomap y Laplacian on Matlab.

Performances are assessed using DET plots and measured in terms of equal error rate (EER). The cost function is calculated following NIST criteria [33].

Baseline experiment: speaker recognition experiment trained and tested with spontaneous sentences. Models were obtained from target data, impostors data and test data using GMM-UBM adaptation. CF-dimensional input supervectors are obtained stacking means of the Gaussian components. SVM classifier is trained with target supervectors and used to score the test supervectors.

The three experiments consisted in the performance evaluation of speaker recognition using a new speaker representation in a low dimension space, obtained by the Laplacian Eigenmaps algorithm, of the mean matrix of Gaussian component in a closed set of speakers. These mean matrixes were obtained by GMM-UBM, as baseline. Input supervectors are obtained stacking means of the mixture components in a new space. SVM algorithm was used for classification as baseline.

8.1 Experiment 1. New representation using local information of acoustic classes.

First experiment consists of dividing by regions all initial space where the acoustic classes lie. As we assume that all acoustic classes lie in manifold, then we will divide the original space in *C*-submanifolds where there is a single acoustic class in each *c*-submanifold.

$$M_i = \left\{ \begin{array}{cccc} \underline{\vec{x}_1} & \underline{\vec{x}_2} & \cdots & \underline{\vec{x}_C} \\ \downarrow & \downarrow & & \downarrow \\ x_{1,1} & x_{1,2} & \cdots & x_{1,C} \\ x_{2,1} & \cdots & & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_{F,1} & \cdots & & x_{F,C} \end{array} \right\} \text{ Mean matrix} \qquad S = \{M_1, M_2, \cdots, M_N\} \text{ Speaker Mean matrixes}$$

where $M_i,\ i = 1,2,\cdots,N$ are the mean matrices for N speakers, F is the dimension of each Gaussian component, C is the number of Gaussian components of GMM.

Then, we construct the C subspaces of each Gaussian component for N speakers component.

$$A^c{}_{N,F} = \left\{ \begin{array}{cccc} \underline{1} & \underline{2} & \cdots & \underline{F} \\ \downarrow & \downarrow & & \downarrow \\ M_1(1,c) & M_1(2,c) & \cdots & M_1(F,c) \\ M_2(1,c) & \cdots & & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ M_N(1,c) & \cdots & & M_N(F,c) \end{array} \right\} \qquad c = \{1,2,\cdots,C\} \text{ Gaussian component mixtures.}$$

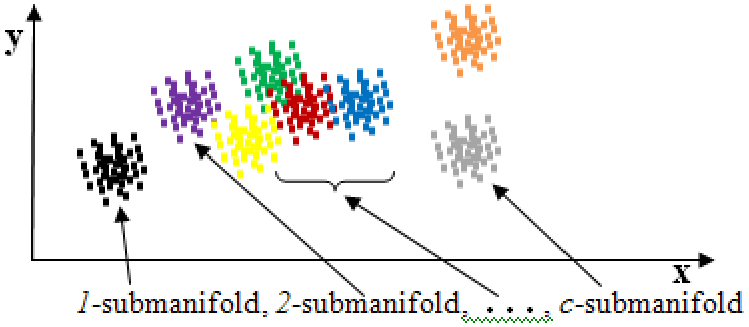


Fig. 7. Acoustic classes regions of several speakers in two dimensions (x,y)

For each submanifold we propose to obtain a new representation in one linear space where the topologic information intervened in the description of this means. In this new space the Gaussian components that belong to the same speaker will be neighbours while the Gaussian components of different speakers must be in different neighbourhoods.

For details of the algorithm go to Algorithm Projections 1.

From this new representation we construct the SV that will participate in the SVM training as well on the SVM test, for each corresponding speaker.

The input parameters were:

Number of neighbors k = 12, number of dimension F = 50, number of Gaussian components C = 128, as a result we will have 6400 dimensions for each speaker supervector. Figure 8 shows DET curve of experiment.

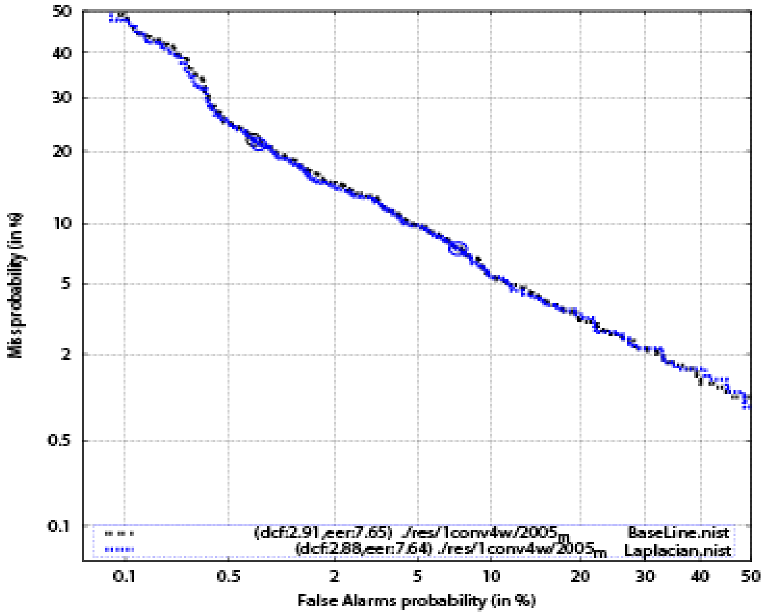


Fig. 8. EER and minDCF performances for BaseLine and using the Laplacian algorithm. Analysis for each submanifold

This experiment, using Laplacian algorithm, reflects very similar results as the baseline, 7.65% ERR (baseline) vs 7.64% EER (Laplacian), 2.91% (baseline) vs 2.88% (Laplacian). This negligible improvement using the Laplacian algorithm is not considered as a good result.

There are several explanations for a not significant improvement in the results:

1. A possible explanation is the difficult that LLE and Laplacian confront with manifolds that contain holes [34]. In addition, LLE and Laplacian tend to collapse large portions ofvery close together data in the low-dimensional space, because the covariance constraint on the solution is too simple [35].
2. Local analysis by submanifolds tends to characterize the local geometry for each Gaussian component and is unable to incorporate information from the rest of the Gaussian distribution, which carry us to a new experiment.
3. It is necessary an analysis of the kernel used in SVM for classification. It is possible that this kernel is not regulated to the new representation.

8.2 Experiment 2. New representation using global information of the acoustic classes.

Second experiment involves a global analysis of all initial space where the acoustic classes lie. As we assume that all the acoustic classes lie in manifold, we want to find some geometric information of the topological structure of the acoustic classes of the speech that better characterizes each speaker.

$$M_i = \left\{\begin{array}{cccc} \underline{\vec{x}_1} & \underline{\vec{x}_2} & \cdots & \underline{\vec{x}_C} \\ \downarrow & \downarrow & & \downarrow \\ x_{1,1} & x_{1,2} & \cdots & x_{1,C} \\ x_{2,1} & \cdots & & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_{F,1} & \cdots & & x_{F,C} \end{array}\right\} \text{ Mean matrix}$$

$$S = \{M_1, M_2, \cdots, M_N\} \text{ Speaker Mean matrixes}$$

where $M_i,\ i = 1,2,\cdots,N$ are the mean matrices of N speakers, F is the dimension of each Gaussian component, C is the number of Gaussian components of GMM.

Then, we construct an initial space for all C Gaussian components of N speakers.

$A_{N,F}^{c}$ = Gaussian component c of all speakers, this step is the same as previous experiment. Then all means of Gausssian components are concatenated into a single matrix that defines each acoustics class, as a result, a new space of all components is obtained.

$$A_{CxN,F} = \left\{\begin{array}{c} A_{N,F}^{1} \\ A_{N,F}^{2} \\ \cdots \\ A_{N,F}^{C} \end{array}\right\}$$

$A_{N,F}^{c}$ are concatenate into a single column of F-dimensions.

where N is the number of speakers, C is the number of Gaussian components and F is the dimension.

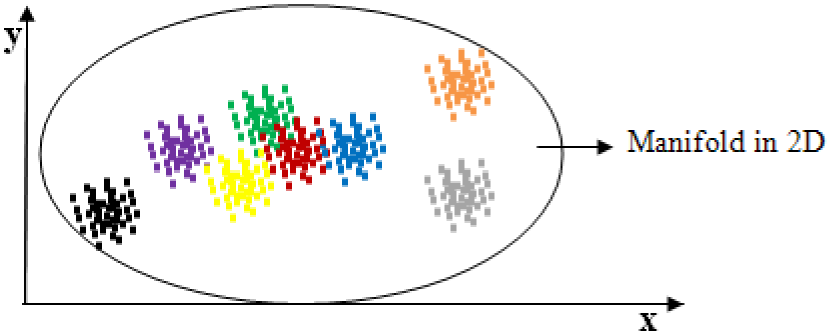


Fig. 9. Manifold where all the acoustic classes lie in two dimensions (x,y)

Then, our proposal was to build a manifold that captures the all inner geometric structure of the mean of the Gaussian components to obtain a new representation in one linear space where the topologic information intervened in the description of this mean. In this space the Gaussian components that belong to the same speaker will be neighbors while the Gaussian components of different speakers must be in different neighborhoods.

For details of the algorithm go to Algorithm Projections 2.

From this new representation we construct the SV that will participate in the SVM training as well on the SVM test, for each corresponding speaker.

The input parameters were:

Number of neighbors k = 12, number of dimension F = 50, number of Gaussian components C = 128, as a result we will have 6400-dimensions for each supervector. Figure 10 shows DET curve of experiment.

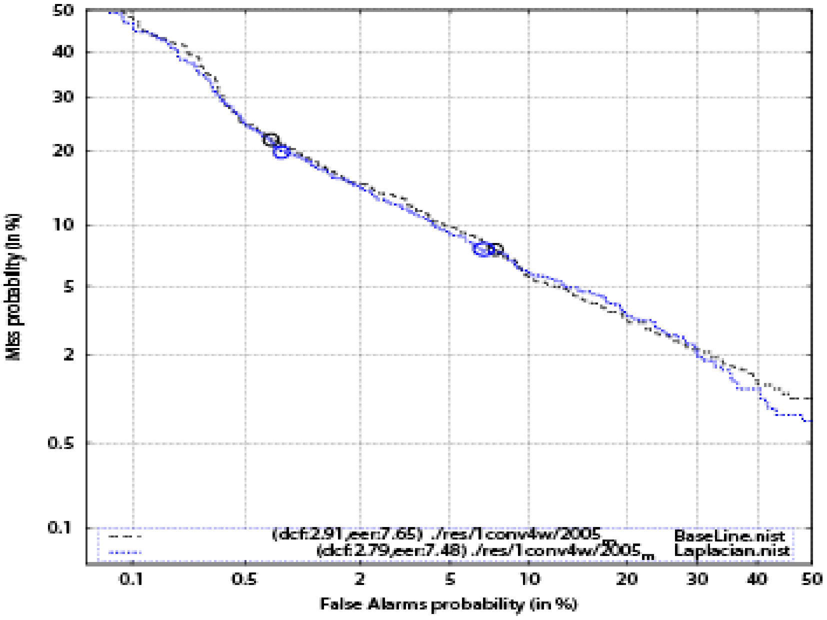


Fig. 10. EER and minDCF performances for BaseLine and using the Laplacian algorithm. Global analysis

This experiment, using Laplacian algorithm, reflects very similar results as the baseline, 7.65% ERR (baseline) vs 7.48% EER (Laplacian) and 2.91% DCF (baseline) vs 2.79% DCF (Laplacian). This negligible improvement using the Laplacian algorithm is not considered as a good result.

This experiment suffers from many of the same weaknesses that previous experiment.

We believe that a more deeply study on this space must be done in the future, especially using the Isomap algorithm which is not used in this experiment by its computational cost and the impossibility at this moment to obtain a rigorous mapping function for this algorithm.

Note that experiments maintain the same dimension in the results, this is because in these experiments we focused only on improving EER and not to reduce the dimensions of the data. The third experiment can be focused to reduce the size of the data.

8.3 Experiment 3 New representation obtained by reducing the acoustic components.

Third experiment involves a global analysis of all initial space where the acoustic classes lie, but at the same time reducing dimension. Otherwise, we assume that all the acoustic classes lie in manifold and we want to find the geometric information of the topological structure of these acoustic classes that better characterizes the speaker, but we will work assuming that each point in this space will be defined by the number of Gaussian components

$$M_i = \left\{\begin{array}{cccc} \underline{\vec{x}_1} & \underline{\vec{x}_2} & \cdots & \underline{\vec{x}_C} \\ \downarrow & \downarrow & & \downarrow \\ x_{1,1} & x_{1,2} & \cdots & x_{1,C} \\ x_{2,1} & \cdots & & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_{F,1} & \cdots & & x_{F,C} \end{array}\right\} \text{ Mean matrix} \qquad S = \{M_1, M_2, \cdots, M_N\} \text{ Speaker Mean matrixes}$$

where M_i, $i = 1,2,\cdots,N$ are the mean matrices of *N* speakers, *F* is the dimension of each Gaussian component, *C* is the number of Gaussian components of GMM.

Then, we construct an initial space for C Gaussian components of all speakers for its F dimension.

$$A^f{}_{N,C} = \left\{\begin{array}{cccc} \underline{1} & \underline{2} & \cdots & \underline{C} \\ \downarrow & \downarrow & & \downarrow \\ M_1(f,1) & M_1(f,2) & \cdots & M_1(f,C) \\ M_2(f,1) & \cdots & & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ M_N(f,1) & \cdots & & M_N(f,C) \end{array}\right\} \qquad A^f{}_{N,C} \text{ matrix built for the dimension } f$$

each matrix $A^f{}_{N,C}$ will have a N x C dimension (number of speaker by number of Gaussian components), and $f = 1,2,\cdots,F$ will be the amount of spaces constructed. For each *f* we make a reduction [2] taking into account the topologic information in each space.

For each submanifold $A^f{}_{N,C}$ we propose to obtain a new representation in one linear space $G : R^C \to R^D$ where the topologic information will intervene in the description of this means. For this, we use Isomap and Laplacian Eigenmaps algorithms to obtain a new projection for each matriz, $D(A^f{}_{N,C}) = A^f{}_{N,D}$, where G is the algorithm used and D represents the new smaller dimension, leading to a reduction in the number of Gaussian components for each model D << C.

Later, each speaker models matrix is reassembled from the new space, and DF-dimensional input SV is obtained stacking the vectors for each new speaker matrix.These SV will participate in the SVM training as well on the SVM test, for each corresponding speaker.

For details of the algorithm go to Algorithm Projections 3.

As a result, dimensionality reduction facilitates, not only, classification, but also compression of high-dimensional data.

The DET curves of speaker recognition experiments are shown in figure 11 and 12, the parameters of the first experiment were:

Baseline, C = 512 Gaussian components, F = 50 dimension and FC = 25600 dimensions for each speaker supervector.

[2] The intrinsic dimensionality of data is the minimum number of parameters needed to account for the observed properties of the data.

Isomap reduction, number of neighbors k = 12, number of dimension F = 50, number of Gaussian components C = 512, as a result we will have D = 128 Gaussian components with the same F-dimension and FD = 6400-dimensions for each speaker supervector.

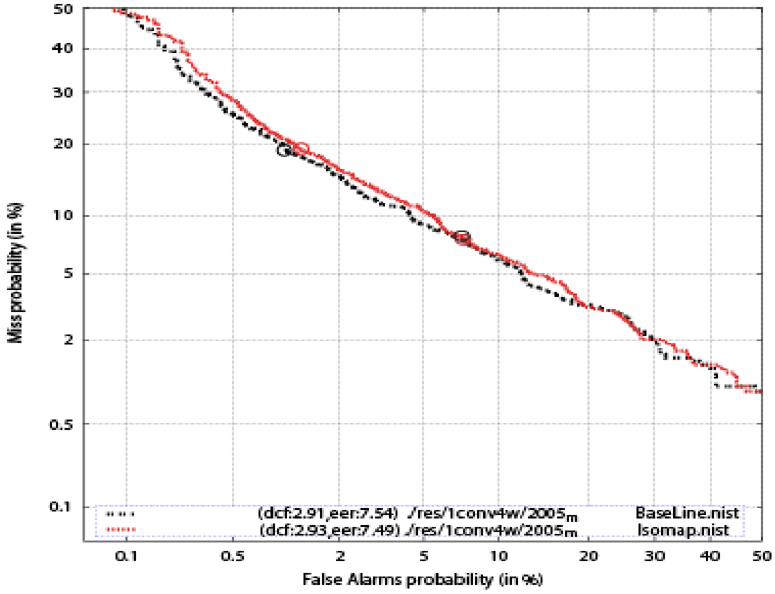


Fig 11. EER and minDCF performances for BaseLine and using the Isomap algorithm

The parameters of the second experiment were:

Baseline, C = 128 Gaussian components, F = 50 dimension and FC = 6400 dimensions for each speaker supervector.

PCA reduction, number of dimension F = 50, number of Gaussian components C = 128, as a result we will have C = 64 Gaussian components with the same dimension and FD =3200-dimensions for each speaker supervector.

Laplacian reduction, number of neighbors k = 12, number of dimension F = 50, number of Gaussian components C = 128, as a result we will have D= 64 Gaussian components with the same F-dimension and FD = 3200-dimensions for each speaker supervector. Fig. 12 shows DET curve of experiment.

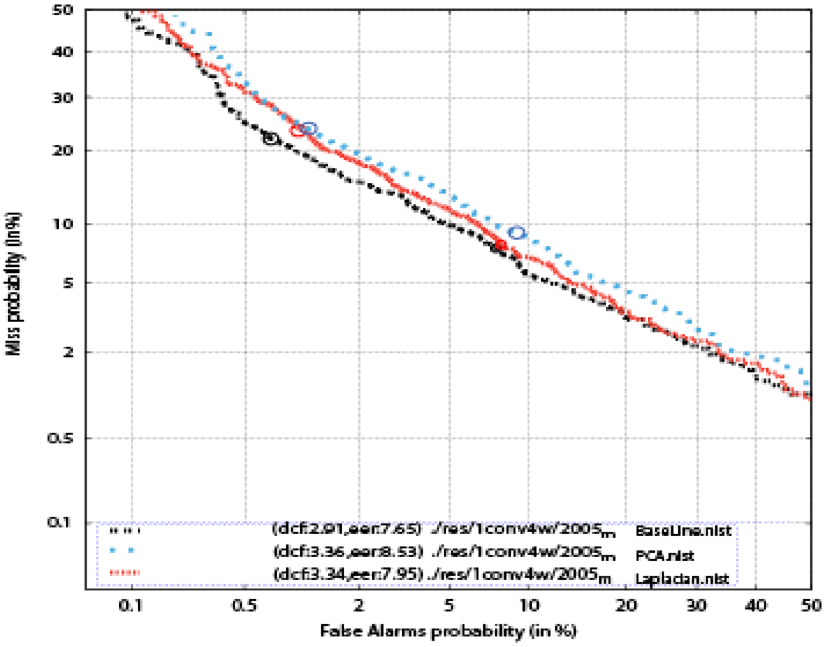


Fig. 12. EER and minDCF performances for BaseLine and using the PCA and Laplacian algorithm

First experiment using Isomap algorithm, reflects similar results as the baseline, 7.54% ERR (baseline) vs 7.49% EER (Isomap) and 2.91% DCF (baseline) vs 2.93% DCF (Isomap).

Second experiment using Laplacian algorithm, reflects very similar results as the baseline, 7.65% ERR (baseline) vs 7.95% EER (Laplacian) and 2.91% DCF (baseline) vs 3.34% DCF (Laplacian). The PCA shows the worst results of the experiment, 8.53% EER and 3.36 DCF.

Not an improvement in EER and DCF were obtained, but in this experiment we focus on reducing the dimension of the supervectors, which were managed successfully showing the large amount of redundant information that exists in the original supervectors. If we compare the SV dimensions in the first experiment, baseline has 25600 dimensions and experiment has 6400 dimensions that is 1/4 of the original size, which is a significant reduction keeping the same EER. In the second, baseline has 6400 dimensions and experiment has 3200 dimensions that are half the size, which is a significant reduction too, although in this experiment, the EER is 0.3 % worse.

9 Future work

We proposed to continue working in the experiments 1 and 2 but using Isomap algorithm with a small group of speakers, reducing the computational cost of the algorithm which could do not affect the accomplishment of the results and could achieve an optimum data representation.

In our future work we will use the means of the GMM-UBM adapted by phonetic segments in one signal for each speaker.

a) Split the speech signal or the MFCC feature matrix by phonetic segments and train a model for each segment using GMM-UBM adapting.

b) Then combine the Gaussian components of all models of the same signal to obtain a set of components for each mixture. This combination achieves the capture, for each set of this mixture, of the dynamic behavior of the signal.
c) Using the topological tools for projecting on a new space each set of components, where is contained the dynamic behavior information and the topological structure information.
d) Perform this process for each speaker and follow a similar process to the baseline for classification.

References

[1] Reynolds D. A.: Speaker identification and verification using Gaussian mixture speaker models. Speech Communication, Vol. 17, pp. 91-108, (August), 1995.
[2] Gauvain J. L. and Lee C.-H.: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans. Speech Audio Process. 2, pp. 291-298, 1994.
[3] Reynolds, D., Quatieri, T., Dunn, R., 2000. Speaker verification using adapted gaussian mixture models. Digital Signal Process. 10 (1), 19–41.
[4] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in Proc. 5th European Conference on Speech Communication and Technology (Eurospeech '97), vol. 2, pp. 963–966, Rhodes, Greece, September 1997.
[5] Fauve, B., Matrouf, D., Scheffer, N., Bonastre, J.-F., Mason, J., 2007. State-of-the-art performance in text-independent speaker verification through open-source software. IEEE Trans. Audio, Speech Language Process. 15 (7), 1960–1968.
[6] A. Martin and M. Przybocki, "The NIST Speaker Recognition Evaluation Series," National Institute of Standards and Technology [Online]. Available: http://www.nist.gov/speech/tests/spk
[7] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in Proc. ICASSP, 2005, pp. 629–632.
[8] P.Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Factor analysis simplified," in Proc. ICASSP, 2005, pp. 637–640.
[9] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms, Tech. Report CRIM-06/08-13," 2005. [Online]. Available: http://www.crim.ca/perso/patrick.kenny/
[10] E. Jon, D. Kim, and N. Kim, "EMAP-based speaker adaptation with robust correlation estimation," in Proc. ICASSP, Salt Lake City,UT, May 2001.
[11] D. Kim and N. Kim, "Online adaptation of continuous density hidden Markov models based on speaker space model evolution," in Proc. ICSLP, Denver, CO, Sep. 2002.
[12] Vapnik V N. Statistical Learning Theory. New York, USA: Wiley, 1998.
[13] Campbell W M. Generalized linear discriminant sequence kernels for speaker recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. Orlando, USA, 2002: 161-164.
[14] Wan V, Renals S. Speaker verification using sequence discriminant support vector machines. IEEE Trans. Speech and Audio Processing, 2005, 13(2): 203-210.
[15] Andrew Errity and John McKenna, "An Investigation of Manifold Learning for Speech Analysis". Interspeech, Pittsburgh, PA, USA. September 17-21, 2006.
[16] S. T. Roweis and L. K. Saul: "Nonlinear dimensionality reduction by locally linear embedding". Science, vol. 290, no. 5500, pp. 2323–2326, December 2000.
[17] L. K. Saul and S. T. Roweis: "Think globally, fit locally: unsupervised learning of low dimensional manifolds". Journal of Machine Learning Research, vol. 4, pp. 119–155, 2003.
[18] J. B. Tenenbaum, V. de Silva, and J. C. Langford: "A global geometric framework for nonlinear dimensionality reduction". Science, vol. 290, pp. 2319–2323, 2000.
[19] M. Belkin and P. Niyogi: "Laplacian eigenmaps and spectral techniques for embedding and clustering". Advances in Neural Information Processing Systems, vol. 14, pp. 585–591, MIT Press, 2002.
[20] Aren Jansen and Partha Niyogi: "Intrinsic Fourier analysis of the manifold of speech sounds". Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, 2006.
[21] Aren Jansen and Partha Niyogi: "A Geometric Perspective on Speech Sounds". Tech. Report TR-2005-08. Computer Science Dept., Univ. of Chicago, 2005.

[22] Aren Jansen: “The Manifold Nature of Vowel Sounds”. Master’s Paper, Dept. of Computer Science, Univ. of Chicago. September 14, 2007.

[23] Douglas A. Reynolds y Richard C. Rose: “Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models”. IEEE Trans. On Speech and Audio Processing, vol. 3, no 1, 1995.

[24] I. Borg and P. Groenen: “Modern Multidimensional Scaling: Theory and Applications”. Springer, 1997.

[25] Christopher J. C. Burges: “Geometric Methods for Feature Extraction and Dimensional Reduction - A Guided Tour”. The Data Mining and Knowledge Discovery Handbook, pp 59-92, ISBN 0-387-24435-2, Springer, 2005.

[26] Mikhail Belkin and Partha Niyogi: “Laplacian eigenmaps for dimensionality reduction and data representation”. Neural Computation, 15(6):1373–1396, June 2003.

[27] Campbell J.P.: “Speaker Recognition: A Tutorial”. Proceedings of the IEEE, vol. 85, no.9, pp.1437-1462, 1997.

[28] Ortega-Garcia, Javier; Gonzalez-Rodriguez, Joaquin; Marrero-Aguiar, Victoria: “AHUMADA A large speech corpus in Spanish for speaker characterization and identification”. Speech Communication, vol. 31, pp 255-264, 2000.

[29] Mikhail Belkin and Partha Niyogi: “Using Manifold Structure for Partially Labelled Classification”. Advances in Neural Information Processing Systems, 15, 2002.

[30] G. Hernandez, J. R. Calvo, F. J. Reyes and R. Fernández: “Simple Noise robust feature vector selection method for speaker recognition”. LNCS vol 5856 pp 313-320 ISBN: 978-3-642-10267-7, 2009.

[31] ALIZE: Open Tool for Speaker Recognition [Online]. Available: http:// www.lia.univ-avignon.fr/heberges/ALIZE/

[32] J.-F. Bonastre, F. Wils, and S. Meignier, “ALIZE, a free toolkit for speaker recognition,” in Proc. ICASSP, 2005, pp. 737–740.

[33] A. Martin and M. Przybocki, “NIST speaker recognition evaluation chronicles,” in Proc. Odyssey, 2004, pp. 15–22.

[34] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by Locally Linear Embedding. Science, 290(5500):2323–2326, 2000.

[35] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality reduction: A comparative review, 2008.

Anexo

The proposed algorithm 1 to obtain new representations is:

Train Algorithm Baseline 1

Input: SV^{i}_{FC} → $i = 1,\ldots,N$ Stacked the Gaussian component mean for each target speakers model (λ_i), where *FxC* is of dimension of the super vectors.

$\widehat{SV}_{FC}$ → Stacked the Gaussian component mean for a set of impostor speakers models, where *FxC* is of dimension of the super vectors.

Output: $ModelSVM_i$ → $i = 1,\ldots,N$, Support vectors of each speaker used in de training.

```
For each i in N
    $ModelSVM_i$ = SVM($SV^{i}_{FC}, \widehat{SV}_{FC}$) → Get support vectors that are between
        $SV^{i}$ and all the impostor set.
End
```

Test Algorithm

Input: SV^{i}_{FC} → $i = 1,\ldots,T$ Stacked the Gaussian component mean for each test speakers model (λ_i), where *FxC* is of dimension of the super vectors.

$ModelSVM_i$ → $i = 1,\ldots,N$, Support vectors of each target speaker

Output: Likelihood → $LLH_{N,T}$

```
For each i in N
    For each j in T
        $LLH_{i,j}$ = SVMtest ($ModelSVM_i, SV^{j}_{FC}$) → Compute the likelihood, get score
            value.
    end
end
```

Algorithm Projections 1

Input: M_i → $i = 1,\ldots,N$ mean matrixes of every speaker (target, impostor and test).

```
Output: ${R^i}_{C,k}{}^{Laplacian}$ → $i = 1,\ldots,N$, $R_i$ is the new mean space for each speaker.
        ${R^i}_{C,k}{}^{Isomap}$ → $i = 1,\ldots,N$, $R_i$ is the new mean space for each speaker.

For each $c$ in $C$
        $A^c_{N,F}$ = Gaussian component $c$ of all speakers $M_i$

        ${P^c}_{N,k}{}^{Laplacian}$ = Laplacian algorithm ( $A^c_{N,F}$ ).      To obtain the projection for
                                                                      each Gaussian component of
        ${P^c}_{N,k}{}^{Isomap}$ = Isomap algorithm ( $A^c_{N,F}$ ).            all speaker, for each algorithm
 end

        ${R^i}_{C,k}{}^{Laplacian}$ =[
                ${P^1}_{i,k}{}^{Laplacian}, \cdots, {P^C}_{i,k}{}^{Laplacian}$ ]
                                                                      For each speaker
        ${R^i}_{C,k}{}^{Isomap}$ =[
                ${P^1}_{i,k}{}^{Isomap}, {P^2}_{i,k}{}^{Isomap}, \cdots, {P^C}_{i,k}{}^{Isomap}$
                ]
```

The proposed algorithm 2 to obtain new representations is:

Algorithm Projections 2

```
Input: $M_i$ → $i = 1,\ldots,N$ mean matrixes of every speaker (target, impostor and test).
Output: $R_{C,k}^{Laplacian}$ → $R$ is the new mean space.
        $R_{C,k}^{Isomap}$ → $R$ is the new mean space.

$A_{NxC,F}$ , $A_{N,k}^{Laplacian}$ , $A_{N,k}^{Isomap}$ →empty matrix, $k$ new dimension. $k << F$.
For each $c$ in $C$
    $A_{N,F}^{c}$ = Gaussian component $c$ of all speakers $M_i$
    $A_{NxC,F}$ = [ $A_{NxC,F}$ ; $A_{N,F}^{c}$ ] . is concatenated into a single column $NxC$ of $F$-dimensions
end
    $P_{NxC,k}^{Laplacian}$ = Laplacian algorithms ( $A_{NxC,F}$ ).    To obtain the projection
    $P_{NxC,k}^{Isomap}$ = Isomap ( $A_{NxC,F}$ ).                      for all Gaussian
                                                                       component of all speaker,
                                                                       for each algorithms
    $P_{NxC,k}^{Laplacian} = P_{NxC,k}^{Laplacian'}$ → transpose
    $P_{NxC,k}^{Isomap} = P_{NxC,k}^{Isomap'}$ → transpose
For each $c$ :$N$: in $NxC$
    $A_{N,k}^{Laplacian} = P_{NxC,k}^{Laplacian}$ ($c$, $c+N-1$) Gaussian component $c$ of all speakers ($X$).
    $A_{N,k}^{Isomap} = P_{NxC,k}^{Isomap}$ ($c$, $c+N-1$) Gaussian component $c$ of all speakers ($X$).
    $R_{C,k}^{Laplacian}$ = [ $R_{C,k}^{Laplacian}$ , $A_{N,k}^{LLE'}$ ] → the components of all speaker were assembled
    $R_{C,k}^{Isomap}$ = [ $R_{C,k}^{Isomap}$ , $A_{N,k}^{LLE'}$ ] → the components of all speaker were assembled
end
```

The proposed algorithm 3 to obtain new representations is:

Algorithm Projections 3

```
Input: $M_i$ → $i = 1,\ldots,N$ mean matrixes of every speaker (target, impostor and test).
Output: $R_{N,Fxk}^{Laplacian}$ → $R$ is the new mean space.
        $R_{N,Fxk}^{Isomap}$ → $R$ is the new mean space.

$R_{N,Fxk}^{Laplacian}$, $R_{N,Fxk}^{Isomap}$, empty matrixes. $k$ new components. k<< C.
For each $f$ in $F$
    $A_{N,C}^{f}$ = the $f$-dimension of each component for all speaker $M_i$
    $P_{N,k}^{Laplacian}$ = Laplacian algorithms ( $A_{N,C}^{f}$ ).    To obtain the projection of the $f$-dimension for all Gaussian component of all speaker, for each algorithms
    $P_{N,k}^{Isomap}$ = Isomap ( $A_{N,C}^{f}$ ).
    $R_{N,Fxk}^{Laplacian}$ = [ $R_{N,Fxk}^{Laplacian}$ , $P_{N,k}^{Laplacian}$ ] are places each new $k$-dimension in the corresponding component.
    $R_{N,Fxk}^{Isomap}$ = [ $R_{N,Fxk}^{Isomap}$ , $P_{N,k}^{Isomap}$ ] are places each new $k$-dimension in the corresponding component.
end
```

RT_035, octubre 2010

Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona

Diseño de Portada: Di. Alejandro Pérez Abraham

RNPS No. 2142

ISSN 2072-6287

Indicaciones para los Autores:

Seguir la plantilla que aparece en www.cenatav.co.cu

CENATAV
7ma. No. 21812 e/218 y 222, Rpto. Siboney, Playa;
Ciudad de La Habana. Cuba. C.P. 12200

Impreso en Cuba