Salcedo-Sanz, S., Gallardo-Antolin, A., Leiva-Murillo, J. & Bousono-Calzon, C. (2006). Offline speaker segmentation using genetic algorithms and mutual information. *IEEE Transactions on Evolutionary Computation*, 10(2), pp. 175–186.

DOI: 10.1109/tevc.2005.857079

Off-line Speaker Segmentation using Genetic Algorithms and Mutual Information.

Sancho Salcedo-Sanz, Member, IEEE, Ascensión Gallardo-Antolín, José Miguel Leiva-Murillo, Student Member, IEEE, and Carlos Bousoño-Calzón, Member, IEEE

The authors are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911 Leganés-Madrid, Spain.

Abstract

In this paper we present an evolutionary approach to speaker segmentation, an activity that is especially important prior to speaker recognition and audio content analysis tasks. Our approach consists of a Genetic Algorithm (GA) which encodes possible segmentations of an audio record, and a measure of mutual information between the audio data and possible segmentations, which is used as fitness function for the GA. We introduce a compact encoding of the problem into the GA which reduces the length of the GA individuals and improves the GA convergence properties. Our algorithm has been tested on the segmentation of real audio data, and its performance has been compared with several existing algorithms for speaker segmentation, obtaining very good results in all test problems.

Keywords

Speaker segmentation, genetic algorithms, mutual information, unsupervised learning.

I. Introduction

Unsupervised learning is generally associated with the idea of using a collection of raw observations $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, sampled from an unknown distribution $p(\mathbf{x})$ to describe properties of $p(\mathbf{x})$. Unsupervised learning has been useful, among other applications, for classification [1], clustering [2], image segmentation [3] and word segmentation in the audio domain [4]. This paper deals with the problem of audio segmentation.

With the ever increasing number of TV channels and radio stations, many hours of TV and radio broadcasts are collected every year by national heritage institutions and private companies. Apart from the architectural problems underlying the design of databases for storing these data, another crucial problem is information retrieval. In audio data files, information retrieval is normally performed by indexing the audio databases, associating each audio document with a file describing its structure in terms of retrieval keys [5]. To perform full indexing, an essential initial step is to determine which speaker is speaking at a given time. This process is known as "speaker segmentation" of the audio data base.

Speaker segmentation consists of distinguishing the utterance of one speaker from another in an audio document. In addition to indexing, the segmentation of audio databases can be useful for speech recognition purposes [5], speaker verification [6], low bit-rate audio coding, environment and channel change detection or providing interesting additional information such as speaker turn and speaker identities (allowing the automatic indexing and retrieval of all occurrences of a same speaker) [7].

The problem of segmenting an audio record has been tackled recently using distance-based methods [7], [8], [9], and hidden Markov models [10], [11], [12]. The former approach obtains good results in the segmentation of speech databases, but some problems of accuracy in the tests performed have been reported, such as the missed detection of short segments. In addition, its performance relies on the selection of a certain threshold which has to be empirically tuned according to the audio record characteristics. The latter method has the drawback that hidden Markov models need to be trained, so a previously labelled training database, or an initial segmentation of the database, is needed. In this paper, we propose an approach that solves these two drawbacks.

Specifically, we consider the problem of the segmentation of audio records containing two speakers. The problem consists of automatically marking the periods of time in which each speaker is talking (speaker turns). We propose an approach to this task which uses an unsupervised learning algorithm, formed by a Genetic Algorithm (GA) for maximizing a measure of Mutual Information (MI) between classes and data. MI is a concept taken from Information Theory [13], which measures the quantity of "common" information between a sequence of labels C and a vector of data $\mathbf{x}$. Intuitively, signals with a high degree of MI between samples and classes are more easily separable than others that contain a lower level. In this paper we use a novel approach to MI [14], which is based on direct approximation of entropy. The samples, $\mathbf{x}$, involved in the calculation of $I(\mathbf{x}, C)$ are the Mel Frequency Cepstrum Coefficients (MFCC) of the audio record, whereas the sequence of classes, C consists of a sequence of binary values $\{0, 1\}$ representing which speaker is currently talking (each bit represents 10 milliseconds of the audio record). A GA is then used to obtain the sequence of labels C^o which maximizes the MI, $I(\mathbf{x}, C)$. Since the problem consists of segmenting audio records containing only two speakers, a GA with binary representation is suitable for this purpose. Thus, every sequence C is codified in the GA as a binary string (GA individual), and represents a possible segmentation of the audio record. After the evolution of the GA, the best solution found represents the optimal sequence C^o (segmentation of the audio record). Note that alternative approaches can be used for maximizing the MI: simulated annealing seems to be another appealing option, since it could be also implemented with binary encoding and using the compaction factor.

Also variable length genetic algorithms which codify the speaker changes as integer numbers instead of binary strings [15] can be applied to solve the problem. Other approaches like fuzzy coding genetic algorithms [**?**] or Estimation of Distribution Algorithms (EDAs) [**?**] could also be used to solve the problem.

To improve the performance of our algorithm we have introduced several modifications to the basic GA working scheme stated above: note that the time a given speaker is talking can be represented in the GA by a string of all 1s (or 0s). Since a given speaker rarely talks less than one second, it is possible to compact every binary string in the GA by a Compaction Factor (CF). Thus, a bit in the compacted string represents CF bits in the non-compacted string. We run the GA in the space of the compacted strings, but the calculation of the fitness values is performed with non-compacted strings. This process allows having shorter binary strings in the GA, and its performance in convergence time is improved.

The use of compact solutions encodings in GAs is not a new topic. There are several works in the literature where the use of compact representations improves the GA performance on a given problem. For example, in [16] a hybrid GA is used for solving the frequency assignment problem in satellite communications, using a compact representation of the problem first introduced in [17]. In this case, the use of a compact encoding instead of the standard one for this problem, is useful for managing the problem's constraints. This is also the case in [**?**], where a GA for the minimum unserviced allocation problem (MUA) is presented. The authors define an alternative encoding which modifies the search space, improving the performance of the GA proposed in the MUA. Another interesting work which introduces a compact representation in a GA is [18]. This paper presents a GA with a compact solution encoding for the container ship stowage problem. In this case, the use of the compact representation allows a significant reduction of the search space, and thus, the GA is able to find more accurate solutions in less time than using the standard representation for this problem. Note that the idea behind our CF is similar to that in [18]: to obtain a reduction of the search space which makes the problem tractable for the GA.

We have tested our approach in the segmentation of real audio records of different

lengths, involving (i) two male speakers, (ii) two female speakers and (iii) one male and one female speakers. We have compared our approach with some other algorithms for speaker segmentation. First, we compare the results obtained by our GA against the results obtained by the DISTBIC algorithm [7]. In spite of the fact that the DISTBIC algorithm solves a more general problem (it relaxes the constraint that only two speakers are involved in the dialogue, as this paper assumes), it is still one of the best known algorithms for solving the speaker segmentation problem, and a comparison with it can provide some insight into the performance of our algorithm. In the experiments section we show that our approach obtains significant improvements over DISTBIC in all tests problems tackled. In addition, we compare our GA against other unsupervised algorithms which can be used for speaker segmentation, such as a standard clustering algorithm, and a Hidden Markov Model (HMM). The comparison with all these methods shows that the GA proposed in this paper is a competitive method for solving the segmentation of audio records.

This article is structured as follows: In the next section, we give the background needed to follow the rest of the paper, including a brief description of MFCC, and some previous approaches to speaker segmentation that can be found in the literature. Section III introduces the proposed GA. It is subdivided in two subsections: first, the MI measure used as fitness function is described in Section III-A, and second, the modifications for adapting the GA to the speaker segmentation problem are described in Section III-B. Section IV shows the results obtained by our algorithm segmenting real audio records, and they are discussed through comparison with the results obtained by some other approaches to the problem. Finally, Section V concludes.

II. Background

In this section we provide some background needed to follow the rest of the paper. We include the description of the MFCC parameterization of speech signals, some previous approaches to the segmentation of audio records and an overview of the concept of detectability in audio records, which will be used in the experiments section.

A. MFCC parameterization of speech signals

Speech (in general, audio) signals need to be parameterized prior to segmentation. Parameterization consists of the extraction of a set of features from the speech waveform, which must present two main characteristics: they must provide a reasonable and compact representation of the speech signal (usually, in the time-frequency domain) and they must have adequate discrimination capabilities for distinguishing between sounds.

MFCC [19] are the most commonly used Fourier-based parameters in automatic speech recognition and speaker recognition applications. In this case, we have decided to use MFCC as proposed in [8], although there are many other alternatives (see [20] for more details). Usually speaker change detection is used for indexing and retrieving information from spoken documents. In this context, automatic speech recognition is used for extracting textual information from audio records. Thus, the use of the same kind of parameters for both the speaker change detection and speech recognition procedures allows a useful reduction in the computational load and memory requirements. Here, we have used 12 MFCC parameters, extracted at a frame period of 10 ms. Although it is usual to complete the feature vectors with their corresponding first derivatives (the so-called Δ-coefficients), we have not proceeded in this way, following the conclusions extracted by [7], since there is no evidence that Δ-coefficients are statistically significantly better or worse than MFCCs.

The procedure for extracting the MFCC parameters is illustrated in Figure 1. The use of MFCC parameters is fairly widespread in speech analysis, the reader is referred to [19] and [21] for a more detailed explanation on the MFCC generation.

B. Previous approaches to segmentation of audio records

Different approaches for segmentation of audio records have been proposed in recent years. They can be classified in three groups:

• **Energy-based methods**: In this approach, it is assumed that sentences uttered by different speakers in a conversation are delimited by pauses [22]. As a consequence, the segmentation relies on the accuracy of an inter-speaker silence detector, which usually works by measuring the energy of each segment and comparing it to a predefined or adaptively estimated threshold. This technique presents two important drawbacks: first,

the accuracy of the segmentation strongly depends on the choice of the energy threshold, and second, it is not always true that people speak between significant silences.

• **Distance-based methods**: This approach consists of measuring the dissimilarity between two adjacent windows of (parameterized) audio data. Depending on the degree of dissimilarity, the system locates a change mark at the point at which the dissimilarity is maximized. Several dissimilarity measures have been proposed in the literature, such as the generalized likelihood ratio [23], [9] the Kullback-Leibler distance or the Bayesian Information Criterion (BIC) [8], [24]. Again, the main drawback is the presence of a threshold which has to be tuned for each kind of audio database. The DISTBIC algorithm [7], which is one of the algorithms used in this paper for comparison purposes, can be considered as a two-pass segmentation method belonging to this group. In the first pass, the generalized likelihood measure is used for determining the approximate situation of the segment boundaries, while in the second pass, these changing points are refined by applying the BIC criterion. A more detailed description of the DISTBIC algorithm is given in Section IV-C. The clustering approach [10] considered in this paper for comparison also belongs to this group of segmentation algorithms.

• **Model-based segmentation**: In this case, a statistical model (for example a hidden Markov model [12], [10], [11], [25]) is trained for a set of predefined acoustic classes (speech, speaker, background noise, music, telephone speech, etc.). For segmentation purposes, each frame (or various frames) of the audio stream is classified using a maximum likelihood criterion and the segment boundaries are located at the temporal point where a change of acoustic class occurs. The main disadvantages of this method include the need to predefine the number and nature of the acoustic classes and the large quantity of labelled data needed for building the different acoustic models in a supervised manner. This last drawback can be tackled using an initial segmentation of the database, as has been shown in [10].

C. Detectability in speaker segmentation

The performance of the majority of the segmentation algorithms strongly depends on the segment length in the audio record. As reported in [8] and in [7], short speaker turns are more difficult to detect than longer ones. Chen and Gopalakrishnan suggest in [8]

a possible measure of the difficulty for detecting a given speaker change, based on the concept of *detectability*.

Let $T = \{t_i\}$ be the sequence of true speaker turns; the detectability of a certain changing point t_i is defined as:

$$D(t_i) = \min(t_i - t_{i-1} + 1,\ t_{i+1} - t_i + 1), \tag{1}$$

where $(t_i - t_{i-1} + 1)$ is the length of the segment previous to the changing point t_i, and $(t_{i+1} - t_i + 1)$ is the length of the segment following to the changing point. In general, when the detectability is low, the current changing point is more likely to be missed, whereas large values of detectability imply that the changing point is often detected.

III. Genetic Algorithm for Speaker Segmentation

In this paper, the search for the sequence of labels C^o which provides a segmentation of the audio record is performed by a GA. Since we tackle the problem of segmenting audio records with two speakers, a binary representation of the problem seems appropriate. Every sequence of classes C representing a possible segmentation of the audio record is codified by means of a binary string, in which each bit represents 10 ms. of audio (this quantity is determined by the frame period used in the parameter extraction procedure (see Subsection II-A)). That is, with the frame period considered in this paper, every minute of audio is encoded by a binary string of length $l = 6000$ bits. We use a standard GA [26], formed by a population of ξ binary individuals, which evolve by means of the classical genetic operators, selection, crossover and mutation. The fitness function associated which each individual of the GA is a measure of MI. In the following sections we describe the measure of MI used as fitness function, and the reduction of GA individuals length by means of the *Compaction Factor*.

A. Fitness Function: Mutual Information

Since the formulation of Shannon's Information Theory, Mutual Information (MI) has been considered a natural measure of the quantity of information that two (or more) signals have in common. Analytically, MI is expressed as the Kullback-Leibler divergence

between the joint probability density function (pdf) of the signals and the product of the marginal densities [13]:

$$I(\mathbf{u},\mathbf{v}) = D_{KL}\Big(p(\mathbf{u},\mathbf{v})|p(\mathbf{u})p(\mathbf{v})\Big) = \int p(\mathbf{u},\mathbf{v}) \log \frac{p(\mathbf{u},\mathbf{v})}{p(\mathbf{u})p(\mathbf{v})} d\mathbf{u}d\mathbf{v}. \tag{2}$$

The calculation of this integral is not easy since the pdfs involved are not usually available. However, some advantage can be taken from the fact that one of the signals is discrete. Actually, this is the case in either supervised or unsupervised learning problems, in which continuous signals are related to a set of discrete, finite classes.

In terms of entropy, MI can also be expressed as:

$$I(\mathbf{u},\mathbf{v}) = h(\mathbf{u}) - h(\mathbf{u}|\mathbf{v}), \tag{3}$$

where

$$h(\mathbf{u}) = -\int p(\mathbf{u}) \log p(\mathbf{u}) d\mathbf{u}, \tag{4}$$

and $p(\mathbf{u})$ is the pdf of the signal.

In a learning problem, the variables involved are the multidimensional data $\mathbf{x} \in \mathbb{R}^d$ and a discrete and finite set of classes $C \in \{c_1, c_2, \ldots, c_K\}$, that are the patterns to be learned. Thus, Equation 3 may be re-expressed as:

$$\begin{aligned} I(\mathbf{x},C) &= h(\mathbf{x}) - h(\mathbf{x}|C) \\ &= h(\mathbf{x}) - \sum_k p(c_k) h(\mathbf{x}|c_k). \end{aligned} \tag{5}$$

Unfortunately, the problem of estimating the entropy is, in the multidimensional case, extremely difficult. Nevertheless, successful efforts have been carried out for one dimensional signals. The problem of estimating the pdfs can be avoided by directly computing the entropies from statistics of the data. The entropy of a one-dimensional variable may be stated as:

$$h(x) = h(x_{gauss}) - J(x), \tag{6}$$

where x_{gauss} follows a Gaussian distribution with the same variance as x and $J(x)$ is the so-called *negentropy*. This quantity is always positive since a Gaussian random variable is, among all the possible distributions with the same variance, the one with the highest entropy. Two cumulant-based polynomial expansions have been traditionally used for the estimation of the $J(x)$: the Gram-Charlier series and the Edgeworth series [27]. However, the terms of higher degree in the expansions make these approximations very sensitive to outliers and samples coming from the "tails" of the distribution.

Alternative estimations of the negentropy have been successfully used in previous works in independent component analysis. One approximation of $J(x)$ is given by the expression:

$$J(x) = k_1 \left[E\left\{ x \exp\left(\frac{-x^2}{2} \right) \right\} \right]^2 + k_2 \left[E\left\{ \exp\left(\frac{-x^2}{2} \right) \right\} - \sqrt{\frac{1}{2}} \right]^2, \tag{7}$$

where k_1 and k_2 are constants defined by $k_1 = \dfrac{36}{8\sqrt{3}-9}$ and $k_2 = \dfrac{24}{16\sqrt{3}-27}$, respectively [28]. This expression has been proven to be more robust and stable against outliers, providing values for the negentropy quite close to the actual ones [28].

Since this approximation is only defined for one-dimensional signals, a slight modification on the cost function must be applied to make use of it. Instead of the original MI described in Equation 5, the MI between each of the MFCC and C will be computed. Thus the following approximation will be used:

$$I(\mathbf{x}, C) \approx \sum_i I(x_i, C), \tag{8}$$

where x_i stands for a MFCC coefficient. This approximation assumes that the cross-entropy between the components is high, and the MI between them negligible. This is the expression we use as the fitness function for the GA.

B. Reduction of GA individual's length

As mentioned above, every individual in the GA encodes every minute of audio to be segmented by means of a binary string of length $l = 6000$ bits. This implies that the search space will have a size of 2^{6000}. In such a search space the GA will have problems of convergence, obtaining low quality solutions. This situation would be even worse with

larger audio records, for example in an audio record of 10 minutes it would be necessary to use binary strings of $l = 60000$, making it computationally expensive for the GA to converge to a solution.

It is possible to overcome this difficulty by looking at the problem's structure. First of all, note that we have codified a solution with an accuracy (resolution) of 10 ms. That is, we would be able to detect changes in speaker with such accuracy using the representation exposed above. This also means that one second of audio is represented by 100 bits. On the other hand, in a standard audio record, a speaker rarely talks for less than one second. This means that the correct solution will have large strings of all 1s and 0s representing the segmentation of the audio. For example if a speaker talks for three seconds before changing to other speaker, the optimal solution would be a string of 300 1s (0s) before changing to 0s (1s). Thus, it is possible to reduce the length of the GA individuals by compacting a number CF of bits into one. In the new representation CF bits are codified as one bit, so the new length of the GA individuals will be $l' = \frac{l}{CF}$.

In our approach, the GA operates on this new representation l', that reduces the search space and improves GA's convergence. We say then that the GA is being run in its *compacted* form. Note, however, that the calculation of the fitness involves individuals of length l and not l' (because of the audio data length), so every individual in the compacted GA has to be expanded, i.e. every bit is expanded to CF identical bits, for the fitness calculation.

This length reduction of individuals in the GA obviously affects the accuracy of the encoding: using the expanded representation we have an accuracy of 10 ms. for detecting changes of speaker. If we use the compacted form of the GA, the accuracy of segmentation is reduced to $10 \cdot CF$ ms. Thus, if for example an accuracy of one second is acceptable for detecting speaker changes, we could set $CF = 100$, and the length of individuals in the GA will be reduced as $l' = \frac{l}{100}$. If we want a higher accuracy, the compaction factor CF has to be smaller.

IV. Simulations and results

In this section, first, we briefly describe the speech databases used in the simulations. Secondly, we describe the assessment measures considered and the different algorithms

implemented for comparison purposes. Finally, we report and discuss the results obtained by our algorithm.

A. Test problems

Two different types of speech data have been used to test the performance of our algorithm: artificially created audio records and real audio records from TV interviews.

• 50 conversations involving 76 different speakers, with a total duration of approximately 62.20 minutes, were artificially created by concatenating sentences from the Resource Management RM1 Database [29]. This database consists of speech recorded at 16 kHz in clean conditions, and it has been widely used by the speech technology community for automatic continuous speech recognition assessment. The original pauses between sentences were shortened to an average duration of approximately 190 ms. for a better simulation of real conversations. The conversations created contain a total of 1071 speaker turns and the duration of each segment varies from 1.05 seconds to 7.25 seconds with an average length of 3.33 seconds. Figure 2 (a) shows the percentage of segments with a given detectability in the artificial data used. Note that the percentage of short turns with a detectability less than 2 seconds is over 14%, the percentage of speaker turns with a detectability between 2 and 3 seconds is about 50% and the changing points with a detectability more than 3 seconds is about 36% . In these experiments, we have divided these conversations into three groups according to the different types of speakers involved: male-male, female-female and male-female. Table I shows the main characteristics of these problems, numbered as #1, #2 and #3. A CF of 20 (which corresponds to a segmentation resolution of 200 ms.) was used in each case.

• A total of 35 TV news broadcasts (corresponding to interviews) with a duration of 55.80 minutes were extracted from the 1997 HUB English Evaluation Speech Database, distributed by NIST [30]. The conversations involve 36 different speakers in this case. The original aim of this database was to foster research on the problem of accurately transcribing broadcast news speech, in which the first step is the segmentation of the speech data into homogeneous segments (same speaker, same acoustic environment). The selected data contains spontaneous speech recorded at 16 kHz and at different acoustic conditions

(clean and in telephone environment). NIST provides hand-segmentations of this data that we have used as a reference. The conversations extracted from this database contain 128 segment boundaries, which correspond to an average segment length of approximately 20.54 seconds, with a maximum length of 73.21 seconds and a minimum of 0.75 seconds. Figure 2 (b) shows the percentage of segments depending on its corresponding detectability. In this case, the percentage of short speaker turns with a detectability less than 2 seconds is about 33%, whereas 67% of the segments have a detectability of more than 2 seconds. The average duration of the pauses between speech segments is about 210 ms. This length distribution is typical for interviews, in which the shortest segments usually correspond to the questions of journalists. Note the differences in detectability (Figures 2 (a) and (b)) between real and artificial audio records. Again, different types of speakers have been considered in the conversations, male-male, female-female and male-female speakers. Table I shows the main characteristics of these problems (#4, #5, #6). Note that in this case the CF used was 30 (which corresponds to a segmentation accuracy of 300 ms.), due to these audio records being longer than the artificial ones.

B. Assessment measures

We distinguish between two type of errors related to speaker turn detection. False alarms or Type-I errors occur when a speaker turn is detected although it does not exist. The false alarm rate (FAR) is defined as:

$$\text{FAR} = 100 \cdot \frac{\text{number of FA}}{\text{number of actual speaker turns} + \text{number of FA}} \ \%. \tag{9}$$

Missed detections or Type-II errors occur when the process does not detect an existing speaker turn. The missed detection rate (MDR) is calculated as follows:

$$\text{MDR} = 100 \cdot \frac{\text{number of MD}}{\text{number of actual speaker turns}} \ \%. \tag{10}$$

In our context, a missed detection is more severe than a false alarm, see [7].

Some authors [11], [9], use two different measures (precision (PRC) and recall (RCL)) which are closely related to FA and MD rates. They are defined as,

$$\text{PRC} = 100 \times \frac{\text{number of correctly found speaker turns}}{\text{number of hypothesized speaker turns}} \%, \tag{11}$$

$$\text{RCL} = 100 \times \frac{\text{number of correctly found speaker turns}}{\text{number of actual speaker turns}} \%. \tag{12}$$

As it is difficult to compare the performance of different algorithms examining FAR-MDR or PRC-RCL pairs, a new metric referred to as the F-measure is frequently used [11], [9]. It is computed as a function of precision and recall measures as follows,

$$\text{F} = \frac{2.0 * \text{PRC} * \text{RCL}}{\text{PRC} + \text{RCL}}. \tag{13}$$

F-measure values fall between 0 and 1. Algorithms achieving a F-measure close to 1 show the best performance.

To compute these different metrics, it is necessary to take into account that the position of the speaker turns are not exactly defined, due to the presence of inter-speaker silences or non-speech sounds [11]. Therefore, it is considered that a changing point is correctly located if it belongs to a time interval $[t_o - \Delta t, t_o + \Delta t]$ in which t_o is the reference mark and Δt is the tolerance (600 ms., in our case).

In the experiments described below, we will indicate FAR, MDR, PRC, RCL and F-measure achieved by our algorithm segmenting the audio files referred above, and we use these parameters for comparing the performance of our algorithm with DISTBIC, clustering and HMM segmentation methods.

C. Algorithms for comparison purposes

C.1 DISTBIC

DISTBIC algorithm [7] is based on the Bayesian Information Criterion (BIC), first proposed in [31]. BIC uses a likelihood ratio, in which it is decided whether two fragments belong to the same source or to two different ones. The log-likelihood ratio associated

with the frame i is defined as:

$$R(i) = \log \frac{L(H_0)}{L(H_1)L(H_2)}, \tag{14}$$

where H_0 is the hypothesis of that there is not a change of source in i. $L(H_0)$ is its corresponding likelihood when a Gaussian distribution is assumed. H_1 assumes all frames with index $\leq i$ to belong to speaker 1 and so H_2 does with index $> i$ and speaker 2.

BIC criterion takes also into account the complexity of the solution. The cost function is given by:

$$\Delta BIC(i, m) = -R(i) + \lambda P(m), \tag{15}$$

where $P(m)$ is the penalizing term when m parameters are used, with λ being a threshold parameter. Samples with the higher ΔBIC are the most likely to correspond to a change.

DISTBIC is based on an sliding-windowing that applies BIC to frames all along the sequence. After measuring the $\Delta BIC(i, m)$, DISTBIC carries out two later steps of refinement and validation that improve the performance obtained if just the BIC criterion were applied.

There are two main parameters (apart from the threshold parameter λ) DISTBIC depends on. The first one is the size of the sliding window from which the Gaussian models are built, i.e. the number of samples with index $\leq i$ the hypothesis H_1 is built according with. This size is usually maintained fixed, with a typical value of 1 second [7]. The second parameter is the shift of the window, which determines the resolution of the method.

C.2 Clustering-based segmentation

Speaker segmentation of audio data files can be carried out by using a group average hierarchical agglomerative clustering algorithm as proposed in [10]. This technique consists of dividing the audio data into a certain number of segments (clusters) and iteratively merging two clusters according to a predetermined metric. As we know that all the audio records considered contain two speakers, this procedure finishes when two clusters (each one containing the part of speech uttered by each speaker) are obtained. Note that the information about the number of speakers in each audio file is also used in the HMM-based (see next subsection) and GA approaches.

In the initialization stage of the clustering algorithm, the data are divided into segments of equal length. The initial size of the clusters determines the resolution of the segmentation procedure and, in this sense, it plays a similar role that the CF factor in the GA approach. Thus, for allowing a fairer comparison to the GA method, the clusters have an initial size of CF in both databases: in the RM1 database, initial clusters consists of 200 ms. of speech and in the HUB 97 database, they consist of speech of 300 ms. length.

The distance between two clusters is based on the log-likelihood ratio defined in Equation 14. For the computation of corresponding likelihoods, each cluster is modelled by tied mixtures of multivariate Gaussian distributions in the cepstral space, which are trained following the procedure described in [10]. We have carried out different experiments varying the number of mixtures and we find that, for our databases, the best results are obtained using 32 mixtures.

C.3 HMM-based segmentation

Hidden Markov models can also be used for speaker segmentation as it has been shown in [10] and more recently in [11] and [25].

In this case, speakers are considered different acoustic classes. Each of these classes is statistically represented by a mixture of multivariate Gaussian densities which are trained before the segmentation. Then, the audio data is classified using a Maximum Likelihood criterion with a Viterbi decoder [32] that yields a set of boundaries between classes corresponding to the hypothesized speaker turns.

For building the corresponding models, some labelled data is needed. As in real applications, it is probably difficult to obtain this audio data, we have adopted an unsupervised strategy in the training stage as proposed in [10]. In particular, we have used the segmentation provided by the agglomerative clustering method described in the previous subsection for the initialization of the speaker models which are adequately retrained using the well known Baum-Welch algorithm [32].

For designing the HMM-based segmenter we have used the HMM topology shown in Figure 3. A similar approach has been proposed in [12] for speech and music segmentation and recently adapted for speaker segmentation purposes in [25]. As in Figure 3, the system consists of two fully connected HMM sub-networks, each one corresponding to

each speaker. Both sub-networks are fully connected in order to allow transitions from one speaker to another, and vice-versa. Internally, each sub-network is composed of several left-to-right connected states associated with the same mixture Gaussian distribution. Self-loops are only allowed in the last state. The number of concatenated states imposes a minimum segment duration and determines the resolution of the algorithm, whereas the self-transition of the last state makes possible to increase the segment duration as much as necessary. For a better comparison with clustering and GA approaches, we have enforced the same constraint of minimum duration: 200 ms. for the RM1 database and (which corresponds to 20 internal HMM states) and 300 ms. (30 internal HMM states) for the HUB 97 database.

As information about prior probabilities of speakers is not available, we have assumed that both speakers are equally likely. Transition probabilities between speakers have been empirically selected in order to favor remaining in the current state (speaker 1 or speaker 2).

In our case, preliminary experiments showed that using 32 mixture components per internal state provides a good segmentation accuracy, so we have used this value in the experiments described in next subsection.

C.4 Comments on the compared algorithms

First of all, note that the DISTBIC is the most general algorithm considered, in the sense that it is able to detect more than two speakers. On the other hand, DISTBIC only detects changes between speakers, without identifying which one are involved in the change. Also, the DISTBIC algorithm depends on a threshold which must be tuned in each database. The GA in this paper only considers the segmentation of files containing two speakers. This is also the case of the clustering and HMM approaches in the implementation considered in this paper. The clustering algorithm uses the information about the number of speakers as stopping criterion. It also uses the same distance measure as the DISTBIC algorithm. The HMM approach starts from the segmentation provided by the clustering algorithm in order to initialize the corresponding acoustic models. Note that in this sense, the HMM algorithm is expected to perform better than the clustering and DISTBIC algorithm. Note also that the GA only uses the mutual information between MFCC and classes for guiding

the search, without any kind of initialization.

D. Results

A conventional GA [26] is used in the simulations, with the MI described in Section III-A as the fitness function, a population of $\xi = 50$ individuals, probability of crossover $P_c = 0.6$, probability of mutation $P_m = 0.01$, and maximum generations equal to 1000. A compaction factor $CF = 20$ has been used in simulations with RM1 database whereas $CF = 30$ has been used in tests with NIST HUB 97 database, as can be seen in Table I. These selections of CF allow a balance between accuracy and length reduction of the GA individuals in the problems considered.

We compare results from our new algorithm with those obtained using an implementation of DISTBIC, the clustering method and the HMM-based algorithm described in Section IV-C. As was mentioned in Section IV-C, DISTBIC is a distance-based segmentation method. Therefore, its performance strongly depends on the choice of the threshold of speaker turn detection (Equation 15). A small value of this threshold produces an over-segmentation (an increase in the false alarm rate); on the contrary, a large value produces an under-segmentation of the data (an increase of the missed detection rate). In order to perform a detailed analysis of DISTBIC algorithm, we have carried out a set of experiments varying this threshold. Then, the corresponding Detection Error Trade-off (DET) curves have been obtained. DET curves show the relationship between MD and FA rates as the DISTBIC threshold varies. Note that the GA, the clustering and the HMM-based algorithm will produce a single point in the DET curves. In addition, the DISTBIC algorithm also depends on the value of the shift of the window parameter. We have conducted two different sets of experiments, the first one with a shift of 100 ms., and the second one with a shift of 200 ms. for the RM1 database, and 300 for the HUB 97 database. These values are comparable with the GA using a CF of 20 (200 ms.) and 30 (300 ms.) respectively. The clustering and the HMM-based algorithms have also been tested with the same resolution.

Figure 4 (a) shows the DET curves obtained with the DISTBIC algorithm in the artificially created audio records, from RM1 database, and the GA, clustering and HMM results, as the average over all the changing points in the database. We have enhanced

the so-called *Equal-Error Rate* (EER) point in the DET curves of DISTBIC. The EER is defined as the point at which false alarms equals missed detections. Note that the closer to the bottom left hand corner is the point obtained by the algorithm, the better is its performance in the segmentation problem. Note that the result obtained by the GA is below the DET curves of DISTBIC, and clustering and HMM-based points. This means that, for a given false alarm rate, the GA always obtains a lower value of missed detections than the other algorithms, and vice versa, given a value of missed detections rate, the corresponding value of false alarm rate is always lower using the GA than using the other approaches.

Figure 4 (b) shows the DET curves, and the GA, clustering and HMM results for the real conversations in the NIST HUB 97 database. It is easy to see that the GA also obtains in this case a solution below the DISTBIC DET curves, and its results is below the points obtained by the clustering and the HMM algorithms. Note that the EER points for the DISTBIC are obtained using a different value of the threshold, which depends on the database, and also on the shift of the window.

To further analyze the performance of our approach, we also present the results obtained by the GA, DISTBIC algorithm (EER point), clustering and HMM-based approaches when there are different types of speakers involved in the conversation: male-male, female-female and male-female are the considered cases. Tables II and III show the different values of FAR, MDR, PRC, RCL and F-measure obtained by our algorithm compared with the results obtained by the other approaches considered (best results are highlighted in boldface). Note that these tables detail the results given in average in Figures 4.

The results in the RM1 (Table II) show that our GA obtains better results than other approaches to the segmentation problem. Note that our GA obtains on average (over all the changing points) better results than the other approaches, in all the measures considered. In problem #3 DISTBIC with a shift of 100 ms. obtains a better result in terms of FAR and PRC but the MDR and the RCL measures are in both cases much better using the GA approach. In all cases, however, the result of F-measure obtained by the GA is better than the one obtained by the other approaches considered.

The results in the HUB 97 database (Table III) show that the GA obtains better results

than DISTBIC in all the cases, but in problem #5, where DISTBIC obtains better results in terms of FAR and PRC, GA is able to obtain better values of MDR, RCL and F-measure. The clustering algorithm and the HMM-based approaches are, in this database, able to obtain better results than the GA in problem #4 (male-female). The GA obtains better results than the clustering and HMM-based algorithms in the rest of the cases using this database. Summarizing, the GA obtains better results considering the average of all speaker changes in this database.

Figure 5 compares the performance of our algorithm only with the DISTBIC algorithm (100 ms.) in a conversation involving two female speakers in NIST HUB 97 database (included in Problem #6). Vertical lines mark speaker turn. Note that our approach is more accurate at detecting speaker changes than DISTBIC in this particular problem. In this figure it is possible to see that most of missed detections produced by DISTBIC are due to short sentences, whereas our approach is able to accurately detect them. Figure 6 shows two examples of the GA convergence in conversations of problems #5 (a) and #6 (b), respectively. The fitness of the best individual in the population is displayed. In both examples, the GA obtained the best segmentation about generation 800, with no further improvements in the remaining generations. Note also that the value of MI is completely different from one conversation to the other, depending on the MFCCs that characterize the conversation.

E. Discussion

For the final discussion, first we analyze if the differences in performance between our GA and the other algorithms compared are statistically significant, and after that, we offer some more insight about the GA's performance, by means of analyzing its behavior in problems with different detectability characteristics.

In Table IV we show the values of a two-tailed z-test [**?**] performed on the differences between our GA and the other approaches considered, for RM1 and HUB 97 databases. We have performed the z-test using the average values of FAR and MDR measures. Values marked with a † are significant at $\alpha = 0.05$. Note that the differences between our GA and all the other algorithms are statistically significant in the RM1 database. For the HUB 97 database, in FAR our GA is better than the DISTBIC with shift of 300 ms. and

clustering, but there is not a statistically significant difference in performance with the DISTBIC with a shift of 100 ms. and the HMM approach. However, our GA performs statistically better than all the other compared approaches in MDR.

Experiments carried out have demonstrated that the approach proposed in this paper provides very good results in the segmentation of audio records. We are interested then in analyzing the behavior of the algorithms in problems with different detectability characteristics. In Table I it is possible to check the detectability of the problems considered. Related to this, we would like to study the accuracy of our algorithm detecting short speaker turns compared with the accuracy of the other algorithms. To study this, we have used the following expression

$$\frac{\text{Number of MD } (D(t_i) < 2\text{s})}{\text{Total number of segments } (D(t_i) < 2\text{s})} \tag{16}$$

This formula measures the amount of missed detections of short speaker turns (detectability under 2 seconds) over the total of short turns with a detectability under 2 seconds. Table V shows the percentage of missed detection of short speaker term for the RM1 and HUB 97 databases and all the algorithms considered. We found that, for the RM1 database, the average percentage of missed detections for all the database using our GA is 6.89%, lower than the value obtained by the other algorithms. In the HUB 97 database, our GA missed only 12.31% of short speaker turns, whereas the DISTBIC (shift= 100 ms.) algorithm missed 48.3% of them. The results of the clustering and HMM approaches are more accurate than the DISTBIC's, however, they are still worse than the obtained by the GA.

V. Conclusions and future work

In this paper we have presented an evolutionary technique for solving the problem of speaker segmentation in an audio record, which is a preparatory step in speaker recognition. We have proposed a GA which encodes possible segmentations, and a measure of the mutual information (MI) between the samples of audio and the individuals of the GA, which is used as GA fitness function. The performance of the GA is improved by introducing a more compact encoding of the GA, by means of the so-called Compaction

Factor, which reduces the search space size and improves the convergence performance of the algorithm. The performance of our approach has been tested and discussed in real audio records, and compared with several existing algorithms for the segmentation of audio records, obtaining very good results in all audio records tested.

Regarding the future research starting from this paper, we plan to extend the genetic algorithm presented to the segmentation of conversations containing more than two speakers. Several adaptations in the GA and in the measure of MI would be necessary in order to adapt our algorithm to that problem.

Acknowledgement

The authors would like to thank Prof. Xin Yao and the anonymous reviewers for their constructive comments and suggestions which have significantly improved this paper.

References

[1] P. Lingras, "Unsupervised rough set classification using GAs," *J. Intell. Inform. Systems*, vol. 16, pp. 215–228, 2001.

[2] L. O. Hall, B. Ozyurt, and J. C. Bezdek, "Clustering with a genetically optimized approach," *Pattern Recognition*, vol. 3, no. 2, pp. 103–112, July 1997.

[3] K. Chen and D. Wang, "A dynamically coupled neural oscilator network for image segmentation," *IEEE Transactions on Neural Networks*, vol. 15, pp. 423–439, 2002.

[4] D. Kazakov and S. Manandhar, "Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming," *Machine Learning*, vol. 43, pp. 121–162, 1997.

[5] T. Zhang and C.-C. J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 44, pp. 441–457, May 2001.

[6] K. Chen, "Towards better making a decision in speaker verification," *Pattern Recognition*, vol. 36, pp. 329–346, 2003.

[7] P. Delacourt and C. J. Wellekens, "DISTBIC: a speaker-based segmentation for audio data indexing," *Speech Commun.*, vol. 32, pp. 111–126, 2000.

[8] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. of the DARPA Broadcast News Trancription and Understanding Workshop*, Landsdowne, VA, USA, 1998.

[9] J. Ajmera, I. McCowan, and H. Bourlard, "Robust speaker change detection," *IEEE Signal Processing Letters*, vol. 11, pp. 649–651, 2004.

[10] L. Wilcox, F. Chen, D. Kimber, and V. Balasubramanian, "Segmentation of speech using speaker identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1994, pp. 161–164.

[11] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel, "Strategies for automatic segmentation of audio data," in *Proc. of ICASSP'00*, 2000, pp. 1423–1426.

[12] J. Ajmera, I. McCowan, and H. Bourlard, "Speech/music segmentation using entropy and dynamism features in a HMM classification framework," *Speech Commun.*, vol. 40, pp. 351–363, 2003.

[13] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.

[14] J. M. Leiva-Murillo, R. Santiago-Mozos, S. Salcedo-Sanz, and A. Artés-Rodríguez, "Symbol decision via genetic optimization of mutual information," in *Proc. of the IEEE Real-Time Systems Symposium*, 2003, pp. 51–56.

[15] C. Y. Lee and E. K. Antonsson, "Variable length genomes for evolutionary algorithms," in *Genetic and Evolutionary Computation Conference (GECCO)*, 2000, p. 806.

[16] S. Salcedo-Sanz and C. Bousoño-Calzón, "A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications," *Applied Intelligence*, 2004, in press.

[17] T. Mizuike and Y. Ito, "Optimization of frequency assignment," *IEEE Transactions on Communications*, vol. 37, no. 10, pp. 1031–1041, Oct. 1989.

[18] O. Dubrovsky, G. Levitin, and M. Penn, "A genetic algorithm with a compact solution encoding for the container ship stowage problem," *Journal of Heuristics*, vol. 8, pp. 585–599, 2002.

[19] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[20] A. A. X. Huang and H.-W. Hon, *Spoken language processing: a guide to theory, algorithm and system development*. Prentice-Hall, 2001.

[21] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-time processing of speech signals*. Prentice Hall, 1993.

[22] M. Nishida and Y. Ariki, "Speaker indexing for news articles debates and drama in broadcasted TV programs," in *Proc. of the Speech Recognition Workshop*, 1997, pp. 67–72.

[23] A. G. Adami, S. S. Kajarekar, and H. Hermansky, "A new speaker change detection method for two-speaker segmentation," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, vol. 4, Phoenix, Utah, USA, 2002, pp. 3908–3911.

[24] J. Ferreiros-López and D. P. W. Ellis, "Using acoustic condition clustering to improve acoustic change detection on broadcast news," in *Proc. of the International Conference on Spoken Language Processing (ICSLP'98)*, Sydney, Australia, 1998.

[25] G. Lathoud and I. A. McCowan, "Location based speaker segmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2003, pp. 176–179.

[26] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[27] S. Haykin, *Neural Networks*. Prentice Hall, 1999.

[28] A. Hyvarinen, "New approximations of differential entropy for independent component analysis," in *Advances in Neural Information Processing System*, vol. 10, 1998, pp. 273–279.

[29] *The Resource Management corpus (RM1)*, 1998.

[30] *Linguistic Data Consortium, Catalog No. LDC2002S11*, 1997, http://morph.ldc.upenn.edu/Catalog/LDC2002S11.

[31] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[32] L. R. Rabiner, "A tutorial in hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

List of Tables

List of Figures

TABLE I

MAIN CHARACTERISTICS OF THE TEST PROBLEMS TACKLED

Problem #	Data Base	Duration (minutes)	Percentage of segments with $D < 2$ s	CF	Number of speaker turns	Type of speakers
#1	RM1	21.10	16.24	20	361	male-female
#2	RM1	20.70	9.85	20	382	male-male
#3	RM1	20.41	20.15	20	328	female-female
#4	HUB 97	25.80	50.75	30	52	male-female
#5	HUB 97	21.12	18.35	30	54	male-male
#6	HUB 97	8.90	27.27	30	22	female-female

TABLE II

CHANGE DETECTION RATES (FAR, MDR, PRC, RCL –ALL IN PERCENTAGES–, AND FACTOR F) OBTAINED WITH OUR APPROACH, COMPARED WITH DISTBIC, AGGLOMERATIVE CLUSTERING AND HMM-BASED METHODS FOR THE RM1 DATABASE. THE BEST RESULTS ARE INDICATED IN BOLDFACE.

Experiment	FAR	MDR	PRC	RCL	F
GA					
#1	**4.92**	**6.89**	**96.13**	**93.11**	**0.946**
#2	**2.67**	**2.63**	**98.15**	**97.37**	**0.975**
#3	18.54	**19.03**	76.04	**80.97**	**0.784**
Average	**8.29**	**9.08**	**90.71**	**90.87**	**0.906**
DISTBIC ($\lambda = 2.5$) Shift = 100 ms.					
#1	10.45	8.96	91.31	91.04	0.912
#2	21.54	10.85	72.04	89.15	0.797
#3	**12.14**	27.73	**80.40**	72.27	0.761
Average	14.92	15.38	81.10	84.63	0.825
DISTBIC ($\lambda = 1$) Shift = 200 ms.					
#1	19.23	22.11	77.43	87.89	0.823
#2	28.35	27.89	61.40	72.11	0.663
#3	25.02	24.20	69.56	75.80	0.725
Average	24.26	24.81	69.31	78.57	0.736
CLUSTERING					
#1	11.02	13.87	88.15	86.13	0.871
#2	9.13	7.96	89.06	92.04	0.905
#3	27.89	28.90	66.14	71.10	0.685
Average	15.51	16.37	81.74	83.64	0.826
HMM					
#1	10.95	9.06	89.10	90.94	0.900
#2	9.56	2.96	91.23	97.04	0.940
#3	18.98	25.47	73.45	74.53	0.740
Average	12.91	11.92	85.07	88.56	0.865

TABLE III

CHANGE DETECTION RATES (FAR, MDR, PRC, RCL –ALL IN PERCENTAGES–, AND FACTOR F) OBTAINED WITH OUR APPROACH, COMPARED WITH DISTBIC, AGGLOMERATIVE CLUSTERING AND HMM-BASED METHODS FOR THE NIST HUB 97 EVALUATION DATABASE. THE BEST RESULTS ARE INDICATED IN BOLD-FACE.

Experiment	FAR	MDR	PRC	RCL	F
GA					
#4	25.31	12.24	70.82	87.76	0.783
#5	21.01	**1.32**	79.54	**98.68**	**0.880**
#6	**8.33**	**0.00**	**91.67**	**100.00**	**0.957**
Average	**20.57**	**5.53**	**78.09**	**94.48**	**0.854**
DISTBIC ($\lambda = 3.5$) Shift = 100 ms.					
#4	26.0	35.60	67.81	64.40	0.660
#5	**10.35**	17.93	**90.76**	82.07	0.861
#6	37.14	9.09	60.61	90.91	0.727
Average	21.31	23.37	76.26	76.41	0.756
DISTBIC ($\lambda = 2.7$) Shift = 300 ms.					
#4	40.15	38.96	49.18	61.04	0.548
#5	28.00	29.92	62.23	70.08	0.659
#6	42.11	40.91	44.83	59.09	0.510
Average	35.36	35.48	53.94	64.52	0.588
CLUSTERING					
#4	11.37	16.90	87.51	83.1	0.852
#5	31.84	30.6	59.24	69.4	0.639
#6	58.86	9.09	40.82	90.91	0.563
Average	28.17	21.33	67.56	78.67	0.712
HMM					
#4	**9.12**	**10.01**	**90.36**	**89.99**	**0.917**
#5	32.76	20.91	65.67	79.09	0.717
#6	33.33	13.64	63.33	86.36	0.731
Average	23.25	15.23	75.30	84.77	0.800

TABLE IV

Z-TEST VALUES OF THE STATISTICAL COMPARISON BETWEEN THE GA APPROACH AND THE DISTBIC, CLUSTERING AND HMM METHODS ACCORDING TO THEIR RESPECTIVE PERFORMANCE (FAR AND MDR) FOR THE RM1 AND HUB 97 DATABASES. † STANDS FOR VALUES OF z WHICH ARE SIGNIFICANT AT $\alpha = 0.05$

Experiment RM1	z value (FAR)	z value (MDR)
GA - DISTBIC ($\lambda = 2.5$) Shift = 100 ms.	$5.15^{\dagger}$	$4.47^{\dagger}$
GA - DISTBIC ($\lambda = 1$) Shift = 200 ms.	$11.44^{\dagger}$	$9.92^{\dagger}$
GA - CLUSTERING	$5.56^{\dagger}$	$5.09^{\dagger}$
GA - HMM	$3.69^{\dagger}$	$2.15^{\dagger}$
Experiment HUB 97	z value (FAR)	z value (MDR)
GA - DISTBIC ($\lambda = 2.5$) Shift = 100 ms.	$0,26$	$4,20^{\dagger}$
GA - DISTBIC ($\lambda = 1$) Shift = 200 ms.	$5,50^{\dagger}$	$6,39^{\dagger}$
GA - CLUSTERING	$2,73^{\dagger}$	$3,81^{\dagger}$
GA - HMM	$0,95$	$2,58^{\dagger}$

TABLE V

MISSED DETECTIONS OF SHORT SPEAKER TURNS FOR THE RM1 AND HUB 97 DATABASES

Experiment (RM1)	$\frac{Number\ of\ MD\ \ (D(t_i) < 2s)}{Total\ \ number\ of\ segments\ \ (D(t_i) < 2s)}$
GA	6.89 %
DISTBIC ($\lambda = 2.5$) Shift = 100 ms.	23.54 %
DISTBIC ($\lambda = 1$) Shift = 200 ms.	52.28 %
CLUSTERING	16.29 %
HMM	11.92 %
Experiment (HUB 97)	$\frac{Number\ of\ MD\ \ (D(t_i) < 2s)}{Total\ \ number\ of\ segments\ \ (D(t_i) < 2s)}$
GA	12.31 %
DISTBIC ($\lambda = 3.5$) Shift = 100 ms.	48.30 %
DISTBIC ($\lambda = 2.7$) Shift = 300 ms.	52.74 %
CLUSTERING	32.81 %
HMM	20.76 %

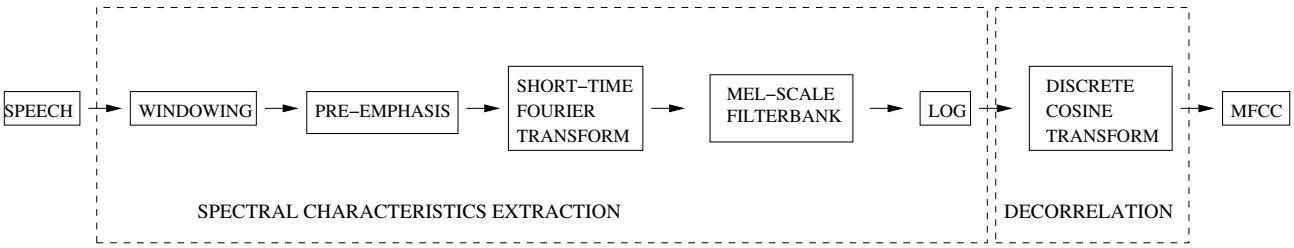
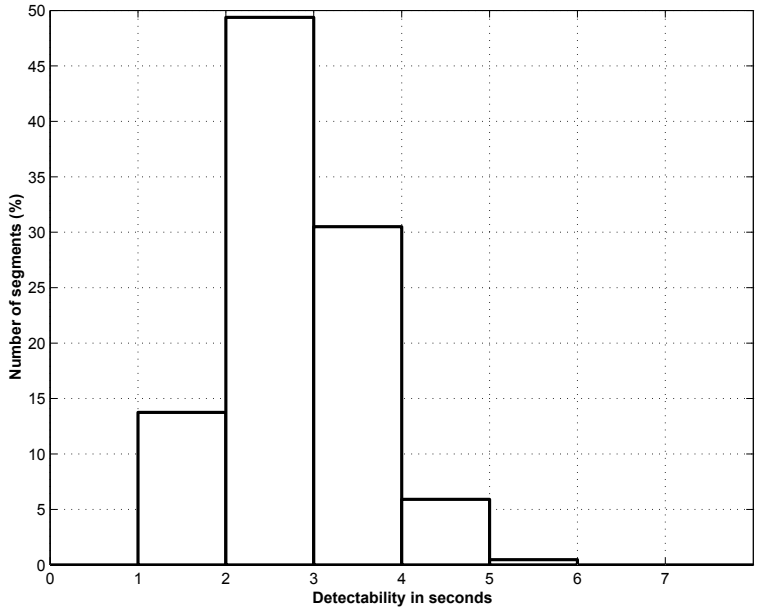
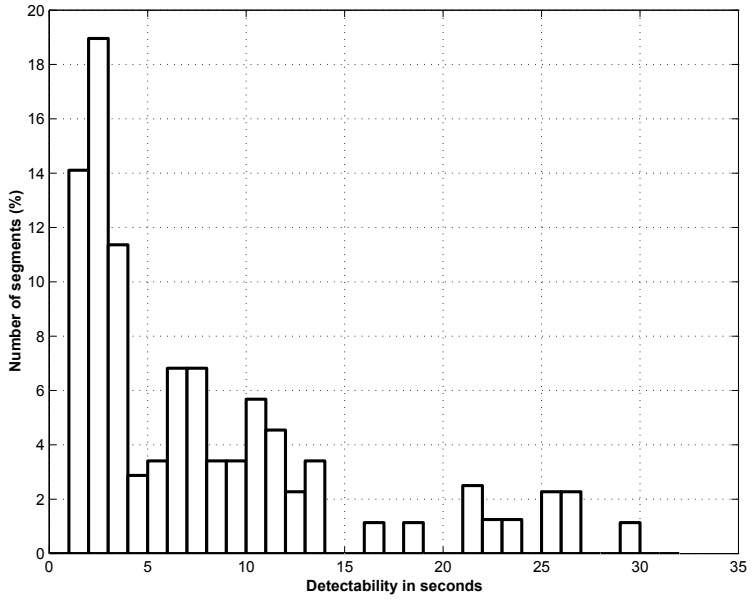


Fig. 1. Top-level block diagram of the speech parameterization procedure. The diagram illustrates the steps followed for the extraction of the MFCC coefficients.



(a)



(b)

Fig. 2. (a) Detectability histogram for RM1 database used; (b) Detectability histogram for NIST HUB 97 database used.

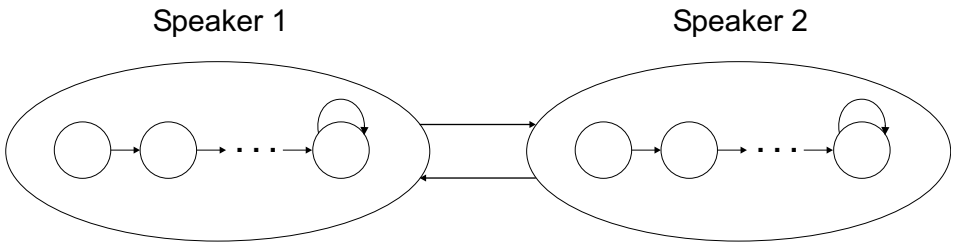
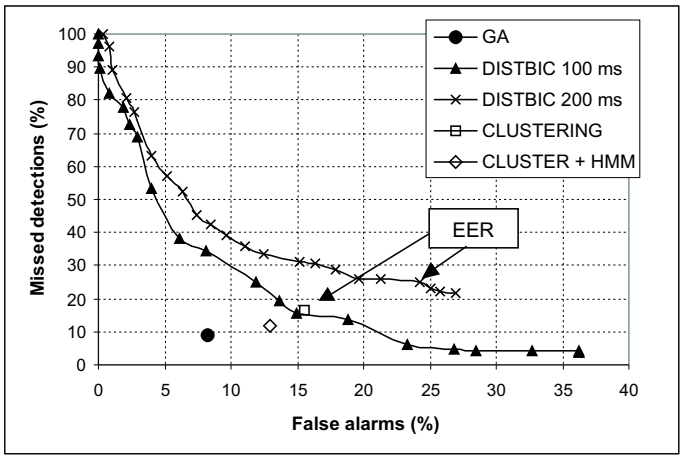


Fig. 3. HMM topology for the HMM-based speaker segmentation system.



(a)

(b)

Fig. 4. (a) DET curve obtained varying the DISTBIC threshold parameter, and FA-MD rates obtained using the GA approach, for RM1 database; (b) DET curve obtained varying the DISTBIC threshold parameter, and FA-MD rates obtained using the GA approach, for NIST HUB 97 database.

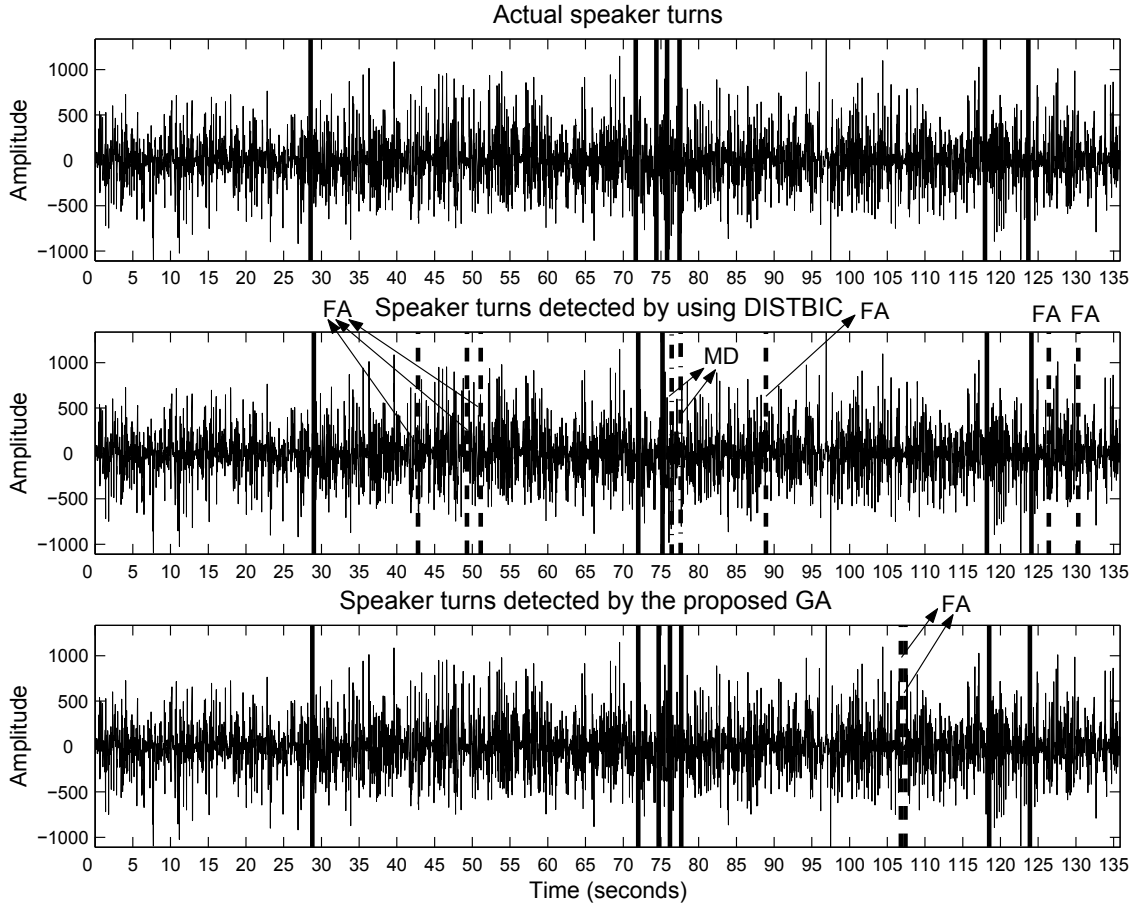
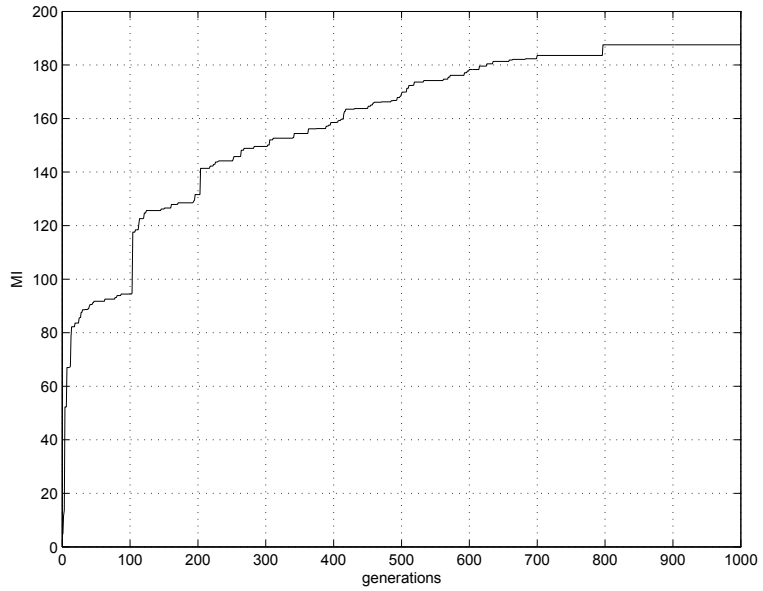
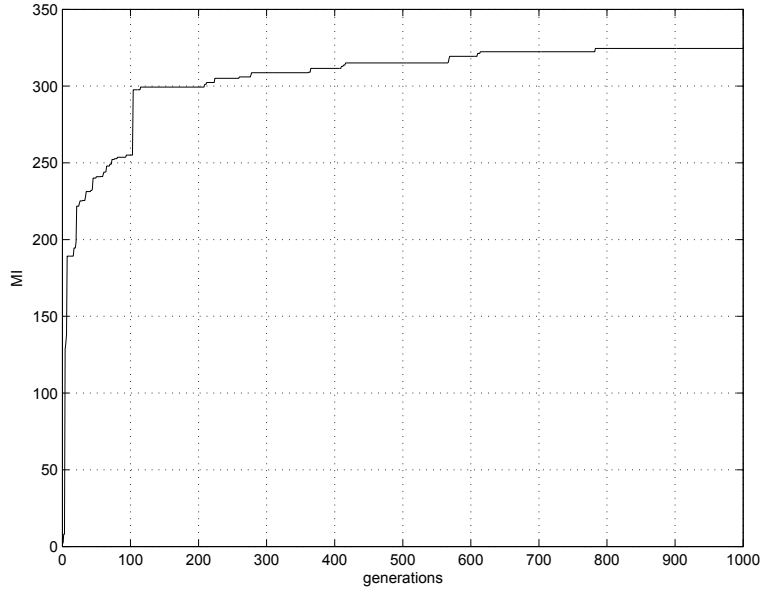


Fig. 5. Actual speaker turns (upper), speaker turns detected by using DISTBIC with a resolution of 100 ms. (medium) and speaker turns detected by the proposed GA (lower) in a conversation from test problem #6. Well-detected speaker boundaries are indicated by solid lines, false alarms (FA) by dashed lines and missed detections (MD) by dotted lines.



(a)



(b)

Fig. 6. (a) GA convergence for one of the conversations of problem #5; (b) GA convergence for one of the conversations of problem #6.