Speaker Segmentation and Clustering

Margarita Kotti, Vassiliki Moschou, Constantine Kotropoulos *

Artificial Intelligence and Information Analysis Lab, Department of Informatics, Aristotle University of Thessaloniki, Box 451, Thessaloniki 54124, Greece, Tel: +30-2310-998225, Fax: +30-2310-998225

Abstract

This survey focuses on two challenging speech processing topics, namely: speaker segmentation and speaker clustering. Speaker segmentation aims at finding speaker change points in an audio stream, whereas speaker clustering aims at grouping speech segments based on speaker characteristics. Model-based, metric-based, and hybrid speaker segmentation algorithms are reviewed. Concerning speaker clustering, deterministic and probabilistic algorithms are examined. A comparative assessment of the reviewed algorithms is undertaken, the algorithm advantages and disadvantages are indicated, insight to the algorithms is offered, and deductions as well as recommendations are given. Rich transcription and movie analysis are candidate applications that benefit from combined speaker segmentation and clustering.

Key words: Speaker segmentation, Speaker clustering, Diarization.

1 Introduction

Nowadays, a rapid increase in the volume of recorded speech is manifested. Indeed, television and audio broadcasting, meeting recordings, and voice mails have become a commonplace [1]. However, the huge volume size hinders content organization, navigation, browsing, and retrieval. *Speaker segmentation* and *speaker clustering* are tools that alleviate the management of huge audio archives.

Speaker segmentation aims at splitting an audio stream into acoustically homogeneous segments, so as every segment ideally contains only one speaker [2]. The MPEG-7 standard developed by the Moving Picture Experts Group can be

* Corresponding author.
Email addresses: mkotti@aiia.csd.auth.gr (Margarita Kotti), vmoshou@aiia.csd.auth.gr (Vassiliki Moschou), costas@aiia.csd.auth.gr (Constantine Kotropoulos).

used to describe efficiently a speech recording [3,4]. For example, MPEG-7 low-level audio feature descriptors such as AudioSpectrumProjection, AudioSpectrumEnvelope [5,6], AudioSpectrumCentroid, AudioWaveformEnvelope [7,8] can be used. MPEG-7 high-level tools, such as SpokenContent, that exploit speakers' word usage or prosodic features, could also be exploited.

Speaker clustering refers to unsupervised classification of speech segments based on speaker voice characteristics [9]. That is, to identify all speech segments uttered by the same speaker in an audio recording and assign a unique label to them [10]. Many speaker clustering methods have been developed, ranging from hierarchical ones, such as the bottom-up (also known as agglomerative) methods and the top-down (also known as divisive) ones, to optimization methods, such as the K-means algorithm and the self-organizing maps [9,11]. Speaker segmentation could precede speaker clustering. However, in such a case the segmentation errors degrade clustering performance. Alternatively, speaker segmentation and clustering can be jointly optimized [12–16].

Speaker segmentation followed by speaker clustering is called *diarization* [2,15,17]. Diarization has received much attention recently, as is manifested by the specific competitions devoted to it under the auspices of the *National Institute of Standards and Technology* (NIST). Diarization is the process of automatically splitting the audio recording into speaker segments and determining which segments are uttered by the same speaker. It is used to answer the question "who spoke when?". Diarization encompasses *speaker verification* and *speaker identification*. In automatic speaker verification, the claimed speaker identity is tested whether it is true or not [18–20]. In automatic speaker identification, no a priori speaker identity claims are made and the system decides who the speaker is.

Several applications of speaker segmentation and speaker clustering could be identified. The first application is *rich transcription* [15,21]. Rich transcription adds several metadata in a spoken document, such as speaker identity, sentence boundaries, and annotations for disfluency. A second application is *movie analysis*. For example, *dialogue detection* determines whether a dialogue occurs in an audio recording or not. Further questions, such as who the interlocutors are or when actors appear, could also be addressed.

Speaker segmentation and clustering are appealing research areas as it is manifested by the numerous research groups and research centers that compete worldwide. Sample representative cases are discussed here. However, the discussion is, by no means, exhaustive. For example, world-wide competitions such as the *segmentation task*, hosted by NIST [22] take place regularly. Segmentation task aims at finding the story boundaries in broadcasts. During benchmark tests, two members of the ELISA consortium, namely the Laboratoire Informatique d'Avignon (LIA) and the Communication Langagiere et Interaction Personne-Systeme (CLIPS), demonstrated an automatic diarization system, which combines two approaches. The first

approach relies on speaker segmentation followed by clustering, while the second one uses an integrated strategy where speaker segmentation and speaker clustering are jointly implemented. Speaker segmentation research at the Center for Spoken Language Research of the Colorado University has been performed as a part of the National Gallery of the Spoken Word project [23]. This project focuses on audio stream phrase recognition in the context of information retrieval. The Global Autonomous Language Exploitation (GALE) program, where the Speech Group at the International Computer Science Institute at Berkeley contributes to, deals with speech recognition, diarization, sentence segmentation, machine translation, and information distillation in various languages [24]. It is seen that GALE partially aims at speaker segmentation and speaker clustering. The Transonic Solutions project, where the Speech Analysis and Interpretation Laboratory (SAIL) at the University of Southern California is active, deals also with speech segmentation as a subproblem of the more challenging speech to speech language translation [25]. Information Extraction from Speech explored by the International Speech Technology and Research (STAR) Laboratory of the Stanford Research Institute (SRI) in California aims at enhancing and integrating speech and natural language processing technology in order to enable information extraction from audio sources. This project, that is funded by the Defense Advanced Research Projects Agency (DARPA), develops several speaker segmentation techniques [26]. Microsoft research has also been active in speaker segmentation, as a part of the Audio Content Analysis project, where discrimination among six audio classes is considered, namely: pure speech, speech over music, speech over noise, pure music, background sound and pause/silence [27]. Robust speaker segmentation has been widely investigated by the Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP) at Switzerland. Another closely related problem studied by IDIAP is location-based multichannel speaker segmentation that is explored within the Hearing Organization And Recognition of Speech in Europe project (HOARSE). Augmented Multi-party Interaction (AMI) is another project undertaken by IDIAP, which is concerned with real-time human interaction, in the context of smart meeting rooms and remote meeting assistants [28]. The Spoken Language Processing Group in the Computer Sciences Laboratory for Mechanics and Engineering Science (LIMSI-CNRS) at Paris has also invested research effort in rich transcription of multilingual spoken documents [29]. The Department of Speech, Music and Hearing of the Royal Institute of Technology (KTH) at Stockholm is interested in speaker segmentation as a preprocessing step of the human-computer interaction task. Such is the case of Computers in the Human Interaction Loop (CHIL) project. Wavesurfer is a publicly available sound manipulation tool developed by KTH, which facilitates segment annotation [30]. Finally, the I6-Aachen group has developed an automatic segmentation algorithm for MPEG audio streams, through the Advisor project, which targets at developing a toolbox for content analysis and rapid video retrieval [31]. Interdisciplinary research across computer science/engineering, cognitive science, and psychology opens additional flourishing directions. One might mention the research related to whether infants temporally segment speech into units within the Infant Speech Segmentation Project at Berkeley University [32]; the research focusing on child-

directed speech, segmentation, and word discovery in the Language Science Research Group at University of Washington [33]; and the research in stuttering as a developmental speech disorder at the Psychology Speech Group of London [34].

The outline of the paper is as follows. Recent advances in speaker segmentation are reviewed in Section 2. This Section includes feature extraction, a taxonomy of speaker segmentation algorithms, figures of merit, representative speaker segmentation algorithms, and a discussion on speaker segmentation algorithms developed by the authors. Section 2 concludes with comparative assessment of the aforementioned speaker segmentation algorithms and an outlook on speaker segmentation algorithms. Section 3 is devoted to speaker clustering. In this Section, evaluation measures, methods for automatic estimation of the number of clusters, deterministic as well as probabilistic approaches to speaker clustering are described. Performance comparisons between the reviewed speaker clustering algorithms are also discussed. Finally, conclusions are drawn in Section 4.

2 Speaker Segmentation

In this Section, firstly feature extraction is briefly studied (subsection 2.1). Next, the problem of speaker segmentation is addressed (subsection 2.2). The Bayesian Information Criterion (BIC) speaker segmentation algorithm is described in detail. In subsection 2.3, commonly used figures of merit for speaker segmentation are defined, their relationships are established, and several state-of-the-art speaker segmentation algorithms are discussed. A comparative study of the reviewed speaker segmentation algorithms is undertaken in subsection 2.4. Discussion, deductions, and insight in speaker segmentation algorithms conclude the Section (subsection 2.5).

2.1 Feature extraction

Let $x[n;m],\ \ n$=$m-N_{sl}+1$,$m-N_{sl}+2$,...,m, define the speech amplitudes of an N_{sl} samples-long audio frame ending at sample m. Let also $w[n;m]$ be the window used to truncate an utterance into frames having typically a duration of 15-20 ms [54]. The most frequently used window is the Hamming window. Hanning window is also widely applied. If $s[n]$ stands for the speech amplitude at sample n, $x[n;m]=s[n]w[n;m]$.

Different features yield a varying performance level [35,42]. *Mel-Frequency Cepstral Coefficients* (MFCCs), sometimes with their first (delta) and/or second (delta-delta) differences are the most common features [2,5,35–46]. *Line spectral pairs* (LSPs) are also widely employed [35,42,48]. *Perceptual linear prediction* (PLP)

cepstral coefficients are utilized in [49]. Other frequently applied features are: short-time energy [15], zero-crossing rate (ZCR) [35], and pitch [35,42,50]. Features based on phoneme duration, speech rate, silence detection, and prosody are investigated in [50]. Silence detection is also used in [51]. Features like the smoothed zero-crossing rate (SZCR), the perceptual minimum variance distortionless response (PMVDR), and the filterbank log-coefficients (FBLCs) are introduced in [53]. Additional features are derived from MPEG-7 audio standard such as AudioSpectrumCentroid, AudioWaveformEnvelope [7,8], AudioSpectrumEnvelope, and AudioSpectrumProjection [5,6].

Definitions of several features are summarized next. It is important to note that there are numerous alternative definitions for certain features (e.g. pitch) to those listed here. Accordingly, the following definitions should be considered as indicative only in such cases. For the features derived from MPEG-7 audio standard whose definitions are omitted due to space constrains, the interested reader may consult [52].

Linear Prediction Coefficients (LPCs)

To simplify notation, the dependence of speech frames on m will be omitted. LPCs are the coefficients $a_\kappa,\ \kappa = 1, \ldots, P$ of an all-pole rational spectrum $\Theta(z)$:

$$\Theta(z) = \frac{G}{1 - \sum_{\kappa=1}^{P} a_\kappa z^{-\kappa}} \tag{1}$$

where G is the model gain. The model gain G can be approximated by estimators employing the coefficients a_κ and the sequence of the autocorrelation coefficients of the speech signal $r_x(\kappa)$ at lags $\kappa = 0, 1, \ldots, P$, by

$$G \simeq \begin{cases} \sqrt{r_x[0] - \sum_{\kappa=1}^{P} r_x[\kappa] a_\kappa} & \text{for unvoiced frames} \\ \sqrt{\frac{1}{F_0}(r_x[0] - \sum_{\kappa=1}^{P} r_x[\kappa] a_\kappa)} & \text{for voiced frames} \end{cases} \tag{2}$$

where F_0 denotes the pitch frequency [55]. In (1), a_κ are obtained as solution of the Yule-Walker equations by means of the Levinson-Durbin algorithm [55,56].

LPC derived cepstrum coefficients

They are defined as:

$$b_\kappa = \begin{cases} \ln G & \kappa = 0 \\ a_\kappa + \sum_{i=1}^{\kappa-1} \frac{i}{\kappa}\, \theta[i]\, a_{\kappa-i} & 0 < \kappa \leq P \\ \sum_{i=\kappa-P}^{\kappa-1} \frac{i}{\kappa}\, \theta[i]\, a_{\kappa-i} & \kappa > P \end{cases} \tag{3}$$

where $\theta[i]$ is the inverse $\mathcal{Z}$-transform of $\Theta(z)$ [55,56].

Line Spectral Pairs (LSPs)

Let $A(z)$ be the polynomial in the denominator of $\Theta(z)$. In (1), $A(z)$ can be decomposed into two $(P+1)$-order polynomials:

$$A_1(z) = A(z) + z^{-(P+1)}A(z^{-1}) \tag{4}$$
$$A_2(z) = A(z) - z^{-(P+1)}A(z^{-1}). \tag{5}$$

The roots of $A_1(z)$ and $A_2(z)$ lie on the unit circle including $\pm$ 1. Once sorted, the roots of $A_1(z)$ and $A_2(z)$ alternate. Besides the pair (1,-1), an LSP is formed by a root of $A_1(z)$ for κ even and a root of $A_2(z)$ for κ odd [55,56].

Mel-Frequency Cepstral Coefficients (MFCCs)

They are computed as follows.

(i) Define a filter bank with P triangular filters,

$$H_p[k] = \begin{cases} 0 & k < f[p-1] \\ \frac{2(k-f[p-1])}{(f[p+1]-f[p-1])(f[p]-f[p-1])} & f[p-1] \leq k \leq f[p] \\ \frac{2(f[p+1]-k)}{(f[p+1]-f[p-1])(f[p+1]-f[p])} & f[p] \leq k \leq f[p+1] \\ 0 & k > f[p+1] \end{cases} \tag{6}$$

whose center frequencies $f[p]$ with $p = 1, \ldots, P$ are uniformly spaced in the mel-scale:

$$f[p] = \Big(\frac{N_{sl}}{F_s}\Big) B_{mel}^{-1}\Big(B_{mel}(F_l) + p\frac{B_{mel}(F_h) - B_{mel}(F_l)}{P+1}\Big). \tag{7}$$

In (7), F_s is the sampling frequency in Hz, whereas F_l and F_h are the lowest and the highest frequencies of the filterbank in Hz. Typical values for F_l and F_h are 0 Hz and $F_s/2$ Hz, respectively. The mel-scale B_{mel} is defined by $B_{mel}(F) = 1125\ln(1 + \frac{F}{700})$.

(ii) Compute the log-energy in the output of each filter.

$$S[p] = \ln\Big[\sum_{k=0}^{N_{sl}-1} |X[k;m]|^2 H_p[k]\Big], \quad 0 < p \leq P, \tag{8}$$

where $X[k;m]$ is the short-term Discrete Fourier Transform (DFT) of $x[n;m]$.

(iii) Compute the discrete cosine transform of the P log energies $S[p]$

$$c[n] = \sum_{p=0}^{P-1} S[p] \cos(\pi n \frac{2\,p-1}{2\,P}), 0 \leq n < P \tag{9}$$

P varies for different implementations usually from 24 to 40. [56].

Short-Time Energy (STE)

A convenient representation of the amplitude variation over time is obtained by [15]:

$$STE[m] = \frac{1}{N_{sl}} \sum_{n=m-N_{sl}+1}^{m} x^2[n;m]. \tag{10}$$

Pitch

An algorithm is presented in [58–60] that includes the following steps:

(i) The signal is low filtered at 900 Hz.

(ii) Clipping is applied to each frame

$$\widehat{x}[n;m] = \begin{cases} x[n;m] - C_{thr} & |x[n;m]| > C_{thr} \\ \\ 0 & |x[n;m]| < C_{thr} \end{cases} \tag{11}$$

where C_{thr} is the 30% of the maximum value of $|x[n;m]|$.

(iii) The short-term autocorrelation $r[\eta;m]$ is calculated, where $\eta = 0, 1, \ldots, N_{sl} - 1$ is the lag. For negative lags, the even symmetry of the autocorrelation is applied.

(iv) The pitch is given by:

$$\widehat{F}_0(m) = \frac{F_s}{2\,N_{sl}} \underset{\eta}{\operatorname{argmax}}\, r[\eta;m]_{\eta=N_{sl}(2\,\mathcal{F}_l/F_s)}^{\eta=N_{sl}(2\,\mathcal{F}_h/F_s)} \tag{12}$$

where $\mathcal{F}_l$ is the lowest pitch frequency preserved by human (typically 50 Hz) and $\mathcal{F}_h$ is the highest frequency preserved by human (typically 500 Hz). Typical pitch values range from 65 Hz to 500 Hz and highly depend on whether the speaker is

male or female [61].

AudioSpectrumCentroid (ASC)

It is the center of gravity of the log-frequency power spectrum:

$$ASC = \frac{\sum_{k=0}^{N_{sl}/2} \log_2(\frac{f[k]}{1000})\Gamma_m[k]}{\sum_{k=0}^{N_{sl}/2} \Gamma_m[k]}, \tag{13}$$

where $\Gamma_m[k]$ are the modified power spectrum coefficients and $f[k]$ are their corresponding frequencies in Hz. The term modified power spectrum coefficients means that the power spectrum coefficients corresponding to frequencies below 62.5 Hz are replaced by a single coefficient equal to their sum [7,8,52].

AudioWaveformEnvelope (AWE)

It is pair of contours. The first contour corresponds to the maximum amplitude found in each speech frame. The second one corresponds to the minimum amplitude for each speech frame [7,8,52]. A frame has a typical duration of 10 ms.

Different features can complement each other in different contexts. For example, individual segmentation results obtained by using separately MFCCs, LSPs, and pitch were fused by using a parallel Bayesian Network in [35,42]. In par to the just described approach, the authors employed individually MFCCs, the maximum of the DFT magnitudes, STE, AudioSpectrumCentroid, and AudioWaveformEnvelope for segmentation and the separate segmentation results were fused in a tandem Bayesian Network [7].

2.2 Speaker segmentation algorithms

In principle, energy-based segmentation, that depends on thresholding the short-time energy could be used for speaker segmentation. However, the accuracy of such a naive technique is poor [69]. Accordingly, more advanced speaker segmentation algorithms are needed, that can be broadly classified into three categories: *model-based*, *metric-based*, and *hybrid* (i.e., combined metric- and model-based) ones.

In *model-based segmentation*, a set of models is derived and trained for different speaker classes from a training corpus. The incoming speech stream is classified using these models. As a result, prior knowledge is a prerequisite to initialize the speaker models. Starting from the less complicated case, a universal background model (UBM) is trained off-line to create a generic speaker model [16,39,48]. During segmentation, this model discriminates between speech and non-speech segments. Since models have been pre-calculated, the algorithm can be used in

real-time. The so-called universal gender models (UGM) can be exploited. Another generic model, the sample speaker model (SSM), is a predetermined generic speaker-independent model that is built by sampling the input audio stream [45]. A more complicated technique is the anchor model, where a speaker utterance is projected onto a space of reference speakers [63]. Finally, models can be created by means of hidden Markov models (HMMs) [15,43,64,65] or support vector machines (SVMs) [66,67]. Model-based segmentation algorithms tend to achieve a moderate recall rate at a high precision rate.

Metric-based segmentation assesses the similarity between neighboring analysis windows shifted over the audio stream by a distance function of their contents. The local maxima of the distance function, which exceed a threshold, are considered as change points. The aforementioned analysis windows may overlap or not, depending on the application. Metric-based methods do not require any prior knowledge on the number of speakers, their identities, or the signal characteristics. A wide variety of distance metrics could be used. For example, a weighted squared Euclidean distance is proposed in [38]. A commonly used metric is the Kullback-Leibler divergence [37,42,68] or the Gaussian divergence (also known as symmetric Kullback-Leibler-2 divergence) [16]. Entropy loss has also been applied [69]. Second-order statistics, such as sphericity, have been proposed in [37,70]. The Hotelling T^2 statistic is another closely related metric [44,53,71]. Alternatively, the generalized likelihood ratio (GLR) test can be applied [15,45]. The most popular criterion is BIC [7,8,11,36,37,40,41,44,73–76]. Metric-based segmentation algorithms generally yield a high recall rate at a moderate precision rate.

Next, we describe BIC due to its prominent position in the related literature. BIC, was originally introduced by Chen and Gopalakrishnan [11] and was obtained by thresholding the GLR [40]. It is an asymptotically optimal Bayesian model-selection criterion used to decide which of N_c parametric models represents best M data samples $\mathbf{x}_i \in \mathbf{R}^d,\ i = 1, 2, \ldots, M$. The samples $\mathbf{x}_i$ are simply vectors of dimension d, having as elements the features described in subsection 2.1. $\mathbf{x}_i$ are assumed to be independent. For speaker segmentation, only two different models are employed (i.e. N_c=2). Assuming two neighboring analysis windows X and Y around time t_j, the problem is to decide whether or not a speaker change point occurs at t_j. Let $Z = X \cup Y$. The problem is formulated as a statistical test between two hypotheses. Under H_0 there is no speaker change point at time t_j. The data samples in Z are modeled by a multivariate Gaussian probability density function (pdf) whose parameters are the mean vector and the covariance matrix. Let $\boldsymbol{\theta}_Z$ denote the aforementioned parameters. $\boldsymbol{\theta}_Z$ could be estimated by either maximum likelihood (ML) or employing robust estimators, such as M-estimators [77]. The log-likelihood L_0 is calculated as:

$$L_0 = \sum_{i=1}^{N_X} \log p(\mathbf{x}_i|\boldsymbol{\theta}_Z) + \sum_{i=1}^{N_Y} \log p(\mathbf{y}_i|\boldsymbol{\theta}_Z) \tag{14}$$

where N_X and N_Y are the numbers of data samples in analysis windows X and Y, respectively. Under H_1 a speaker change point occurs at time t_j. The analysis windows X and Y are modeled by distinct multivariate Gaussian densities, whose parameters are denoted by $\boldsymbol{\theta}_X$ and $\boldsymbol{\theta}_Y$, respectively. Then, the log-likelihood L_1 is obtained by:

$$L_1 = \sum_{i=1}^{N_X} \log p(\mathbf{x}_i|\boldsymbol{\theta}_X) + \sum_{i=1}^{N_Y} \log p(\mathbf{y}_i|\boldsymbol{\theta}_Y). \tag{15}$$

The dissimilarity between the two neighboring analysis windows X and Y is estimated by the BIC criterion defined as:

$$\delta = L_1 - L_0 - \frac{\lambda}{2} \underbrace{\left(d + \frac{d(d+1)}{2}\right)}_{\text{model parameters}} \log N_Z \tag{16}$$

where $N_Z = N_X + N_Y$ is the number of samples in analysis window Z. In (16), λ is a data-dependent penalty factor (ideally 1.0). If $\delta > 0$, a local maximum of δ is found and time t_j is considered to be a speaker change point. If $\delta < 0$, there is no speaker change point at time t_j.

Hybrid algorithms combine metric- and model-based techniques. Usually, metric-based segmentation is used initially to pre-segment the input audio signal. The resulting segments are used then to create a set of speaker models. Next, model-based re-segmentation yields a more refined segmentation. In [43], HMMs are combined with BIC. In [78], after having performed an initial BIC segmentation, the acoustic changes that are not found by BIC are detected in a top-down manner, i.e. through a divide and conquer technique. Another interesting hybrid system is introduced in [15] where two systems are combined, namely the LIA system, which is based on HMMs, and the CLIPS system, which performs BIC-based speaker segmentation followed by hierarchical clustering. The aforementioned systems are combined with two different strategies. The first strategy, called hybridization, feeds the results of CLIPs system into the LIA system, whereas the second strategy, named merging, merges preliminary results from LIA and CLIPs system and re-segmentation is performed using the LIA system.

The majority of algorithms surveyed up to this point are applied to single-channel / single-microphone recordings. A segmentation algorithm can take advantage of different qualities in multiple channels. This is the case of a system which segments dialogues between pilots and traffic controllers [79]. The pilots use one channel for their conversations and the traffic controllers utilize a different channel to instruct the pilots. When multiple microphones are used, speaker location could also be exploited for dialogue detection. An algorithm for segmenting meeting recordings in terms of speaker location is presented in [80]. A source localization technique is applied. This is achieved by modeling the space region by a single Gaussian pdf, and then applying K-means. Alternatively, the between-channel timing informa-

tion can also be employed [81], using spectral-domain cross-correlation in order to detect time difference cues between a pair of microphones. However, multichannel speaker segmentation is left outside the scope of the survey.

2.3 Assessment of speaker segmentation algorithms

In this subsection, commonly used figures of merit for the comparative assessment of speaker segmentation algorithms are defined, benchmark algorithms are briefly described, and speaker segmentation algorithms developed by the authors are discussed.

2.3.1 Figures of merit for speaker segmentation

Two pairs of figures of merit are frequently used to assess the performance of a speaker segmentation algorithm.

On the one hand, one may use the false alarm rate (FAR) and the miss detection rate (MDR) defined as [37,78]:

$$FAR = \frac{FA}{GT + FA},\ MDR = \frac{MD}{GT} \tag{17}$$

where FA denotes the number of false alarms, MD the number of miss detections, and GT stands for the actual number of speaker change points, i.e. the ground truth. A false alarm occurs when a speaker change point is detected, although it does not exist. A miss detection occurs when an existing speaker change point is not detected by the algorithm.

On the other hand, one may employ the precision (PRC), recall (RCL), and F_1 measure given by [41,43]:

$$PRC = \frac{CFC}{DET} = \frac{CFC}{CFC+FA},\ RCL = \frac{CFC}{GT} = \frac{CFC}{CFC+MD},\ F_1 = 2\frac{PRC\ RCL}{PRC+RCL} \tag{18}$$

where CFC denotes the number of correctly found changes and $DET = CFC + FA$ is the number of the detected speaker changes. F_1 measure admits a value between 0 and 1. The higher its value is, the better performance is obtained.

Between the pairs (FAR, MDR) and (PRC, RCL) the following relationships hold:

$$MDR = 1 - RCL,\ FAR = \frac{RCL\ FA}{DET\ PRC+RCL\ FA}. \tag{19}$$

2.3.2 *Performance of BIC-based segmentation algorithms*

Subsequently, emphasis is given to BIC-based speaker segmentation methods due to their popularity and efficiency.

One of the first works applying BIC to speaker segmentation is that of Tritschler and Gopinath [36]. In order to improve the algorithm efficiency and allow for real-time implementation, a couple of heuristics are proposed. First, a varying analysis window scheme is employed. In particular,

(1) A small analysis window of M frames is considered (typically M = 100);
(2) If no speaker change point is found in the current analysis window, the new analysis window size is increased by ΔM_i frames;
(3) If no speaker change point is found in the new analysis window, its size becomes $M + \Delta M_i + \varepsilon_i$ frames, where $\varepsilon_i = 2\,\varepsilon_{i+1}$;
(4) Step (3) is repeated until a speaker change point is found or until the analysis window has reached a maximum size.

Exact values for ΔM_i and ε_i are not specified in [36]. This scheme ensures that the analysis window is increased slowly, when its length is small and in a fast manner, when it gets bigger. Secondly, to speed up computations, BIC is applied only to selected time instants. For example, BIC tests are not performed at the borders of each analysis window, since not enough data are available to build accurate Gaussian models there. Moreover, BIC computations at the beginning of large analysis windows are ignored, since they would be repeated several times. The aforementioned heuristics, due to their efficiency, are commonly used by other researchers [41,44,51,73]. 24-order MFCCs are employed. Experimental results are reported for the HUB4 1997 data [82] yielding FAR=9.2% and MDR=24.7% [36]. The low FAR value can be attributed to the heuristics. However, the reported MDR is relatively high. Since the algorithm is designed for real-time applications, a further refinement of segmentation results is not possible. Obviously, the computational cost is reduced compared to that of the conventional BIC [11].

A two-pass segmentation technique, called DISTBIC, is proposed by Delacourt and Wellekens [37]. The first pass uses a distance computation to determine candidate speaker change points. The second pass applies BIC to validate or discard the candidates selected during the first pass. Six metrics are tested in the first pass: GLR, Kullback-Leibler divergence, and four similarity measures derived from second-order statistics. The aforementioned metrics are computed for a pair of sliding analysis windows. Local maxima of these metrics are fed next to the second BIC pass. A local maximum is considered to be significant if $|\mathcal{D}(\max) - \mathcal{D}(\min_r)| > t_D\sigma$ and $|\mathcal{D}(\max) - \mathcal{D}(\min_l)| > t_D\sigma$, where $\mathcal{D}$ stands for the computed distance, σ denotes the standard deviation of distance measures, t_D is a threshold, whereas $\min_r$ and $\min_l$ minima left and right to the maximum, respectively. Concerning the computation time required to find speaker change points, it is a fraction of the

Table 1
Parameter values, FAR, and MDR for DISTBIC applied to five different data sets [37].

data set	λ	analysis window duration (s)	shift (s)	t_D (%)	First pass		Second pass	
					FAR (%)	MDR (%)	FAR (%)	MDR (%)
TIMIT [87]	1.2	1.96	0.7	15	40.3	14.3	28.2	15.6
CNET [37]	1.0	1.96	0.7	15	18.2	15.7	16.9	21.4
INA [37]	1.8	2.00	0.1	50	37.4	9.03	18.5	13.5
SWITCH-BOARD [88]	1.5	2.00	0.1	50	39.0	29.1	25.9	29.1
jt [37]	1.8	2.00	0.1	50	59.0	8.9	23.7	9.4

recording duration. 12-order MFCCs are used. MFCC first-order differences are also considered, but they are discarded, since the authors claim that they deteriorate performance. Five different data sets are used to evaluate DISTBIC: the first consists of artificially created conversations by concatenating sentences from the TIMIT database [87]; the second is CNET in French [37]; the third, called INA contains French broadcast programs [37]; the fourth is SWITCHBOARD [88]; and the last, called jt, contains also French broadcast programmes [37]. The experimental findings are summarized in Table 1. Distance-based segmentation seems to be more sensitive to environmental changes and speaker intonations in the first pass. This explains its high FAR. The reasoning behind the application of the first pass is that it yields long enough chunks (i.e. segments between two successive candidate speaker change points), before the application of BIC, enabling the accurate estimation of the parameters of the underlying Gaussian models. Table 1 reveals that analysis window duration, shift, and t_D are not sensitive to the language of the recordings. For example, the same parameters for analysis window duration, shift, and t_D are used for conversations in American English from TIMIT and conversations in French from CNET. In addition, analysis window duration, shift, and t_D are not language-sensitive for spontaneous conversations. INA and jt contain spontaneous conversations in French, while SWITCHBOARD includes spontaneous conversations in English. When comparing synthetic conversations the performance is deteriorated for CNET synthetic conversations compared to that for TIMIT ones, as it is demonstrated by MDR. This can be attributed to the fact that CNET includes shorter segments than TIMIT. Finally, the difference in efficiency between synthetic and spontaneous conversations can be attributed to the recording conditions. Segmentation algorithms applied to real conversations detect speaker changes together with recording conditions, whereas when they are applied to synthetic conversations they detect only speaker changes.

Table 2
F_1 measure of by applying SA, DA, and CSA for speaker segmentation on IBNC [73].

Algorithm	F_1 measure (%)
SA	88.4
DA	89.4
CSA	89.4

Three algorithms are compared by Cettolo and Vescovi [73]. They explore the idea of shifting a variable-size analysis window as in [37], but they differentiate in the way they implement growing and shifting of variable-size analysis window as well as the computation of BIC related parameters, such as the mean vectors and the covariance matrices. The first algorithm, the sum algorithm (SA), is a simple method that uses only a sum and a square sum of the data samples $\mathbf{x}_i$ in order to save computations needed for covariance matrix estimation. The second algorithm, the distribution algorithm (DA), applies essentially the algorithm proposed in [36], but it encodes the input signal with its cumulative distribution. The third one, the cumulative sum approach (CSA), represents the input signal by cumulative pairs of sums. Test data are derived from the Italian Broadcast News Corpus (IBNC) [84]. 12-order MFCCs and log-energy are extracted from each speech frame. The reported F_1 measure is shown in Table 2. The figures for the F_1 measure in Table 2 show a similar performance level for all three approaches. Although all these methods invest in reducing computational cost, the CSA has the lowest execution time. It combines the assets of SA and DA, since it encodes the input stream not through the distributions, as in DA, but with the sums of the SA algorithm.

A sequential metric-based segmentation method is introduced by Cheng and Wang [40]. Each speaker change point has multiple chances to be detected by different analysis window pairs. By doing so the method is made more robust than the conventional BIC approach [72]. Two alternatives are described: the sequential metric-based approach with one stage and the sequential metric-based approach with two stages. In the sequential metric-based approach with one stage [40], an analysis window of a typical duration of 2 s is applied. If the BIC value δ becomes positive at time t_j, BIC is performed at $t_j \pm 2$ s and the time instant with the maximum BIC value is set as a change point. In the sequential metric-based approach with two stages, an increased analysis window of duration 2+1=3 s is applied, since when more samples are available, the BIC parameters are estimated more accurately [36,37,40,44,78]. The computational cost of the sequential metric-based approach is linear. The computational cost of the conventional BIC is $O(N_Z^2)$ [72]. The algorithm performance is evaluated in the MATBN2002 Mandarin Chinese broadcast news corpus database [85], using 24-order MFCCs. A FAR equal to 20% is reported for an MDR equal to 20% [78]. The same authors have also proposed metric-SEQDAC [78]. First, to pre-segment a long audio stream into shorter segments, metric-based segmentation with long sliding analysis windows

Table 3
Figures of merit for the method in [41] and the conventional BIC on HUB4 1997 [82] for different values of λ.

method	PRC (%)	RCL (%)	F_1 measure (%)
Ajmera [41]	68	65	67
BIC ($\lambda = 1.0$)	22	81	35
BIC ($\lambda = 4.0$)	46	77	58
BIC ($\lambda = 5.0$)	57	74	64
BIC ($\lambda = 6.0$)	66	71	68
BIC ($\lambda = 7.0$)	71	66	68
BIC ($\lambda = 8.0$)	73	60	66

is applied, since it is fast. Next, refinement is achieved by applying sequentially a divide-and-conquer procedure to each segment in order to detect any remaining change points. The divide-and-conquer procedure searches for speaker change points in a top-down manner, instead of searching for speaker change points in a bottom-up manner, as is widely adopted in the previously described methods. This is very efficient for broadcast news recordings, because many change points are detected quickly. The thresholds for both the sequential metric-based approach and the metric-SEQDAC are determined as in [37]. Metric-SEQDAC efficiency is also evaluated using 24-order MFCCs in the MATBN2002. A FAR of 20% is reported for MDR equal to 16%. Concerning metric-SEQDAC, the second step yields MDR improvement.

A criterion that does not require tuning λ in (16) is proposed by Ajmera et al. [41]. The data samples under H_0 are modeled by a Gaussian mixture model (GMM) with two components instead of a single Gaussian density. Let $\boldsymbol{\theta}'_Z$ be the GMM parameters. Then, (16) is modified to

$$\delta' = L_1 - \sum_{i=1}^{N_X} \log p(\mathbf{x}_i|\boldsymbol{\theta}'_Z) + \sum_{i=1}^{N_Y} \log p(\mathbf{y}_i|\boldsymbol{\theta}'_Z). \tag{20}$$

The number of parameters used to model the data in the two hypotheses are forced to be the same, so that the likelihoods are directly comparable. As a result, the authors claim that no tuning is needed and the criterion is expected to be robust to changing data conditions. The performance of (20) against (16), for several values of λ in the latter, is tested on HUB 1997 [82] using 24-order MFCCs. The results are summarized in Table 3. From Table 3, it is clear that λ can admit other values than 1, as is also the case in [36,37]. In fact, λ = 1 is not the ideal case. The best results are measured for $\lambda = 6.0$ and $\lambda = 7.0$. It is easily seen that higher values of λ yield a higher PRC rate and less false alarms.

Table 4
Figures of merit for the method in [43] tested on a television broadcast audio stream.

method	PRC (%)	RCL (%)	F_1 measure (%)
segment-level segmentation	63.33	36.81	45.20
segment-level segmentation and model-level segmentation	72.72	51.53	60.31
segment-level segmentation, model-level segmentation, and HMM-based re-segmentation	86.36	75.41	80.51

A hybrid speaker segmentation algorithm is developed by Kim et al. [43]. The speech stream is segmented in three stages. In the first stage, called "segment-level segmentation", T^2 statistics are calculated for every possible speaker change point of the analysis window Z and their peak value is chosen as a candidate speaker change point. Then, each speaker change point is either confirmed or discarded by BIC. Next, clusters are built by hierarchically merging segments with respect to the difference of the BIC values between the two segments. The second stage is called "model-level segmentation". An HMM is used in order to determine the actual number of speakers. The third stage is "HMM-based re-segmentation". In this stage, the speaker models from each cluster are re-estimated by HMMs. The algorithm is tested on one audio track from a television talk show program using 23-order MFCCs. The reported figures of merit are listed in Table 4. The hybrid algorithm is more efficient than the metric-based algorithm, as is expected [15]. Moreover, the HMM-based re-segmentation step considerably improves the results. It can be considered as a refinement step, complementary to the commonly used pre-segmentation [7,8,37,40,44,78]. Metric-based segmentation is placed prior to the model-based one, since prior knowledge is a prerequisite for model-based segmentation.

The idea of utilizing Hotelling T^2 statistic prior to BIC for unsupervised speaker segmentation is also exploited by Zhou and Hansen [44]. In this case, the Hotelling T^2 statistic is used to pre-select candidate speaker change points, which are then re-evaluated by BIC. Inspired by [36], three improvements are applied. First, a variable-size increasing analysis window analysis scheme is used. Second, BIC tests are not performed near to analysis windows boundaries, since BIC tends to be unreliable when a change point is adjacent to an analysis window boundary. This improvement lowers the MDR for short segments. Third, frame skipping is applied. That is, not all the frames within an analysis window are considered as candidate change points. Frame skipping combined with Hotelling T^2 statistic pre-selecting speeds up the algorithm by a factor of 100 compared to [11]. The dynamic computation of the analysis window mean and covariance matrix is adopted from the DA algorithm [73]. Frame energy, 12-order MFCCs, and their respective first-order differences are extracted from each frame. The rates reported on the HUB4

Table 5
Figures of merit for the method proposed in [51] and the conventional BIC on TDT-3 Mandarin audio corpus [83].

method	$FAR\,(\%)$	$MDR\,(\%)$
MDL-based segmentation	14	10
BIC	30	29

1997 evaluation data [82] are: $FAR = 16.5\%$ and $MDR = 22.6\%$. There are several advantages in using the Hotelling T^2-BIC scheme. First, by pre-selecting the candidate points with respect to the Hotelling T^2 statistic, the computation of two full covariance matrices is avoided and as a result the computational cost is reduced to $O(N_Z d)$, from $O(N_Z^2)$ needed in the conventional BIC [11]. Second, BIC faces problems when the analysis windows are not sufficiently long or when a change point occurs near the analysis window boundary. This is because insufficient data for estimating second-order statistics render BIC unreliable. First-order statistics needed for Hotelling T^2 statistic are more robust than second-order statistics in the small-sample case, as are segments of duration shorter than 2 s [44].

An algorithm for detecting multiple speaker change points in one analysis window, instead of just a single speaker change point, is proposed by Wu and Hsieh [51]. First, silent segments are deleted, then minimum description length (MDL) is employed instead of BIC. MDL is a two-hypothesis GLR criterion, where multiple change points rather than a single one are assumed in H_1 hypothesis. Consequently, multiple Gaussian models are computed. The first change point detected by MDL is the most abrupt change in Z, the second change point is the second most abrupt change in Z, and so on. Hierarchical binary segmentation uses MDL to generate the optimal change point sequence. 12-order MFCCs, their corresponding first-order differences, the logarithmic energy, and the first-order difference of the logarithmic energy are used as features. To accelerate MDL parameter estimation (i.e., mean vector and covariance matrix estimation), a dynamic computation is adopted, as in DA algorithm [73]. A variable-size sliding analysis window strategy is also utilized [36]. Experiments are conducted in parts of the Topic Detection and Tracking-Phase 3 (TDT-3) Mandarin audio corpus from the Linguistic Data Consortium (LDC) [83]. The efficiency of the MDL-based segmentation compared to conventional BIC [72] is demonstrated in Table 5. MDR is improved by 65.5% relatively to that obtained by the conventional BIC. The relative improvement is 53.3% for FAR compared to that of the conventional BIC and it may be attributed to the larger analysis windows. A typical analysis window length in [51] is 40 s. Consequently, more data are available for building more accurate Gaussian models. Moreover, a change point has multiple chances for being detected, since MDL-based segmentation is recursive, whereas a change point in BIC-based segmentation has only one chance for being detected.

2.3.3 Algorithms proposed by the authors

Three distinct systems for speaker segmentation have been proposed. All systems do not assume any prior knowledge of either the number of speakers or their identities. The first system employs a multiple-pass algorithm [7]. Commonly used features in speech recognition may not be suitable for speech segmentation [44]. Motivated by the aforementioned statement, novel features are investigated, such as MPEG-7 based features. In particular, the MPEG-7 AudioSpectrumCentroid and the maximum of MPEG-7 AudioWaveformEnvelope are found to improve performance. In detail, the following features are extracted: 13-order MFCCs, the maximum of DFT magnitude, the STE, the AudioSpectrumCentroid, and the maximum of AudioWaveformEnvelope. A dynamic thresholding for scalar features, such as the maximum of DFT magnitude, the STE, the AudioSpectrumCentroid, and the maximum of AudioWaveformEnvelope is used. A fusion scheme is employed to refine the segmentation results. Fusing the segmentation results obtained from different features within the same segmentation method is also applied in [35,42]. A parallel Bayesian Network is utilized in [35,42], whereas in the case under consideration, a tandem Bayesian Network is employed, due to its efficiency and popularity. Every pass can be seen as a pre-segmentation step for the next pass that aims to reduce the number of false alarms, while maintaining a low number of miss detections. Every speaker is modeled by a multivariate Gaussian pdf. Whenever new information is available, the respective model is updated, as in [42]. This is of great importance, since BIC is more robust for larger analysis windows [36,37,40,44,78]. Experiments are carried out on a data set created by concatenating speakers from the TIMIT database [87]. The reported figures of merit reported are: $PRC = 78.6\%$, $RCL = 70.0\%$, F_1 measure $= 72.0\%$, $FAR = 21.8\%$, $MDR = 30.5\%$ [7]. An advantage of the algorithm is that every pass is independent of the others. As a result, if time efficiency is of great importance, the backmost passes can be pruned at the expense of accuracy. A novel pre-segmentation scheme is applied. Indeed, in [37,44] pre-segmentation is achieved by utilizing other measures in addition to BIC, whereas in [7] additional features are employed.

A second system is developed in [8]. It is built of three modules. The first module pre-segments utterances by extracting: the mean magnitude of the DFT, the first-order differences of AudioWaveformEnvelope, the mean STE, the AudioWaveform-Envelope, and the variance of the first-order differences of the magnitude of the DFT. The features are selected from an initial set of 24 features by a branch-and-bound selection strategy with backtracking. The initial set includes the mean and the variance of the following feature values, their first-order and second-order differences: the magnitude of the DFT, the STE, the AudioWaveformEnvelope, and the maximum of AudioSpectrumCentroid. The segmentation criterion is a combination of arithmetic mean, geometric mean, and harmonic mean of the five selected features. Arithmetic mean, geometric mean, and harmonic mean have been previously employed in speaker identification [70] and in the first DISTBIC pass [37]. In the second module, the candidate speaker change points found by the

first module are re-evaluated. At a first step, the Euclidean distance between two chunks is investigated. At the second step, the Hotelling T^2 statistic is measured. In both cases, the employed features are the 13-order MFCCs and their corresponding first-order differences. This can be considered as a tandem Bayesian Network with two detectors. In the two detector case, the tandem network is dominant [86]. The third module has two stages. In the first stage, BIC is computed in conjunction with 13-order MFCCs. The resulting potential speaker change points are fed to the second stage, where BIC in conjunction with MFCC first-order differences validates the final speaker change point set. It is found that after each module the number of chunks is decreased, because specific potential change points are discarded, since there are found to be false. Thus, the length of chunks becomes larger, enabling more accurate parameter estimation [36,37,40,44,78]. The experiments are carried out on the same data set used in [7]. Experimental results yield $PRC = 49.0\%$, $RCL = 81.2\%$, F_1 measure = 60.7%, $FAR = 45.5\%$ and $MDR = 18.8\%$. False alarms can be removed more easily, for example through clustering [16,17,35,37,42,44,71]. PRC and FAR are associated to the number of false alarms, while RCL and MDR depend on the number of miss detections. This means that PRC and FAR are less cumbersome to remedy than RCL and MDR. This is the reason why the algorithm puts a higher emphasis on MDR.

The third system models the distribution of the speaker utterances duration [89]. In this way, the search is no longer "blind" and exhaustive. Consequently, a considerably less demanding algorithm in time and memory is developed. It is found that the inverse Gaussian fits best the distribution of utterance durations. Moreover, feature selection is applied prior to segmentation, aiming to determine an MFCC subset that is the most discriminative for the speaker segmentation task. The branch and bound search strategy using depth-first search and backtracking is employed. 24 out of 36 MFCCs are selected namely: the 1st, the 3rd-11th, the 13th, the 16th, the 22th-33th, the 35th, and the 36th. Those MFCCs are applied along with their first- and second-order differences. Experiments are carried out on two databases. The first database is created by concatenating speakers from the TIMIT database, not the same concatenation as in [7] or in [37]. The second database is derived from the HUB-4 1997 English Broadcast News Speech. For the first database the reported figures of merit are: $PRC = 67.0\%$, $RCL = 94.9\%$, $F_1 = 77.7\%$, $FAR = 28.9\%$, and $MDR = 5.1\%$, while for the second database they are: $PRC = 63.4\%$, $RCL = 92.2\%$, $F_1 = 73.8\%$, $FAR = 30.9\%$, and $MDR = 7.8\%$.

2.4 Comparative assessment of speaker segmentation algorithms

A comparison of the algorithms discussed in subsections 2.3.2 and 2.3.3 is undertaken next.

It should be noted that the efficiency of a speaker segmentation algorithm depends

Table 6
Comparative study of speaker segmentation algorithms.

Algorithm	Data used	Features	Performance Criteria				
			PRC (%)	RCL (%)	F_1 (%)	FAR (%)	MDR (%)
Tritschler and Gopinath [36]	HUB4 1997 [82]	24-order MFCCs	-	-	-	9.2	24.7
Delacourt and Wellekens [37]	TIMIT [87]	12-order MFCCs	-	-	-	28.2	15.6
	CNET [37]		-	-	-	16.9	21.4
	INA [37]		-	-	-	18.5	13.5
	SWITCH-BOARD [88]		-	-	-	25.9	29.1
	jt [37]		-	-	-	23.7	9.4
Cettolo and Vescovi [73] SA	IBNC [84]	12-order MFCCs and log energy	-	-	88.4	-	-
Cettolo and Vescovi [73] DA			-	-	89.4	-	-
Cettolo and Vescovi [73] CSA			-	-	89.4	-	-
Cheng and Wang [40]	MATBN2002 [85]	24-order MFCCs	-	-	-	20.0	20.0
Cheng and Wang [78]			-	-	-	20.0	16.0
Ajmera et al. [41]	HUB4 1997 [82]	24-order MFCCs	65.0	68.0	67.0	-	-
Kim et al. [43] segment-level segmentation	one audio track from a television talk show program	23-order MFCCs	63.3	36.81	45.20	-	-
Kim et al. [43] segment-level segmentation and model-level segmentation			72.72	51.53	60.31	-	-
Kim et al. [43] segment-level segmentation, model-level segmentation, and HMM-based re-segmentation			83.36	75.41	80.51	-	-
Zhou and Hansen [44]	HUB4 1997 [82]	frame energy, 12-order MFCCs and their first-order differences	-	-	-	16.5	22.6
Wu and Hsieh [51]	parts of the Topic TDT-3 Mandarin audio corpus [83]	12-order MFCCs and their first-order differences, logarithmic energy, and its first-order difference	-	-	-	14.0	10.0
Kotti et al. [7] multiple-pass algorithm	concatenated utterances from speakers of the TIMIT database [87]	13-order MFCCs, maximum of DFT magnitude, STE, AudioSpectrumCentroid, and maximum of AudioWaveformEnvelope	78.0	70.0	72.0	21.8	30.5
Kotti et al. [8] three modules algorithm		13-order MFCCs and their first-order differences, mean magnitude of the DFT, first-order difference of the maximum of AudioWaveformEnvelope, mean STE, maximum of AudioWaveformEnvelope, and variance of the first-order differences of the magnitude of the DFT	49.0	81.2	60.7	45.5	18.8
Kotti et al. [89] modeling the speaker utterance duration and feature selection		selected MFCC subset consisting of the 1st, the 3rd-11th, the 13th, the 16th, the 22th-33th, the 35th, and the 36th MFCC along with their first- and second-order differences	67.0	94.9	77.7	28.9	5.1
Kotti et al. [89] modeling the speaker utterance duration and feature selection	HUB4 1997 [82]	selected MFCC subset consisting of the 1st, the 3rd-11th, the 13th, the 16th, the 22th-33th, the 35th, and the 36th MFCC along with their first- and second-order differences	63.4	92.2	73.8	30.9	7.8

highly on the nature of the data, where it is applied to. Experimental results are reported for different databases. Some databases contain speaker segments less than 2 s, which BIC handles unsatisfactory [7,8,36,37,44], while others contain long homogeneous segments. Moreover, some contain spontaneous conversations, like HUB4 1997, INA, SWITCHBOARD, and MATBN2002, while others comprise of synthetic conversations, created from TIMIT database or CNET. Moreover, not all databases are recorded under the same conditions. Consequently, different sampling rates, bits per sample, and audio formats are used. Finally, researchers do not use the same figures of merit nor the same experimental protocol, which complicates direct comparisons. To remedy the situation the NIST sponsored competitions in several speech domains. Data collections compiled by these competitions are mostly distributed by the LDC [90].

Table 6 summarizes the performance of the algorithms reviewed in subsections 2.3.2-2.3.3. We refer to the published figures of merit and deduce as many qualitative results as possible. By re-implementing algorithms, there is always a danger to select several algorithm parameters in a wrong manner due to insufficient details given in the literature. Thus rates different to those reported may be obtained.

To begin with, different features are employed in each algorithm. It is clear that there is a preference towards MFCCs. However, there is no consensus with respect to the MFCC-order. For example, 24-order MFCCs are applied in [36,40,41,78], 23-order MFCCs are utilized in [43], 13-order MFCCs along with their first-order differences are utilized in [7,8], while 12-order MFCCs along with their first-order differences are employed in [51]. It is interesting that several MFCC orders are investigated in [51] before the 12-order MFCCs along with their first-order differences are chosen. In [89], an effort is made to discover an MFCC subset that is more suitable to detect a speaker change. Moreover, there is no consensus with respect to first-order MFCC differences. While, first-order MFCC differences are claimed to deteriorate efficiency in [37], the use of first-order MFCC differences is found to improve performance in [51]. Additional features are also investigated: in [44] the frame energy is extracted; in [51] the logarithmic energy and its first-order difference is used; the STE, the DFT magnitude, the AudioSpectrumCentroid, and the maximum of AudioWaveformEnvelope are employed in [7,8].

One could conduct a performance comparison among the algorithms evaluated on the same database. HUB4 1997 is the most commonly appeared database in Table 6. A direct comparison is possible between the real-time algorithm in [36] and the combination of Hotelling T^2 statistic and BIC algorithm in [44], because they utilize the same heuristics. MDR in [44] is relatively improved by 8.5% compared to that in [36], while FAR is deteriorated by 79.3% relatively to that obtained in [36]. The improved MDR in [44] can be attributed to the pre-segmentation based on the Hotelling T^2 statistic. The deterioration in FAR may be attributed to the fact that, the objective in [44] is to reduce the computational cost.

Another comparison is that between the non-tuned algorithm in [41] and the algorithm modeling the speaker utterance duration introduced in [89]. In [41], the heuristics introduced in [36] are applied, whereas the aforementioned heuristics are not employed in [89]. Moreover, in [89], a systematic effort is made to reduce the computational cost. This is not the case in [41]. PRC is relatively deteriorated by 2.5% in [89] when compared to [41], while RCL is improved by 35.6%. The RCL improvement can be attributed to the fact that BIC tests are performed when a speaker change point is most probable to occur in [89].

The algorithm combining the Hotelling T^2 statistic and BIC presented in [44] and the speaker utterance duration modeling algorithm proposed in [89] are also tested on HUB4 1997. Both algorithms employ MFCCs. In [44], an effort is made to discover the MFCC order that yields the most accurate speaker segmentation results, while in [89] the authors try to discover an MFCC subset that is the most suitable to detect a speaker change. When comparing the rates reported in [89] to those in [44], a relative improvement of 65.5% in MDR is associated to a double FAR. Commenting on MDR, it can be assumed that the pre-segmentation based on the Hotelling T^2 statistic is less efficient than the modeling of speaker utterance duration.

Concatenated utterances from speakers of the TIMIT database are used in the multiple-pass algorithm [7], the three-module algorithm [8], the DISTBIC algorithm [37], and the algorithm that models speakers utterance durations [89]. Although the concatenated utterances are not the same, FAR is improved by 22.6% in [7] relatively to that in [37] at the expense of doubling the MDR reported in [37]. A relative deterioration of 20.5% of MDR in [8] is found to yield a relative deterioration of FAR by 61.3% in [8] compared to that in [37]. In [7,8], no heuristics are performed to boost performance. For that reason, the algorithms in [7,8] are expected to be more robust.

Comparing the algorithm resorted to modeling the speaker utterance duration in [89] to the DISTBIC algorithm [37], MDR value is relatively improved by 67.307%, while FAR is slightly deteriorated by 2.4%. MDR improvement may result from the fact that the BIC tests are performed when a speaker change point is most probable to occur. The improved FAR in [37] may be attributed to second-order statistics used in pre-segmentation .

Alternatively, comparisons can be made between the algorithms tested on broadcast programmes. Broadcast programmes are included in the HUB4 1997 database, the IBNC corpus, the MATBN2002 database, the TV program audio track utilized in [43], and the TDT-3 Mandarin audio corpus. A rough comparison is feasible among the real-time algorithm in [36], the sequential metric-based algorithm in [40], and metric-SEQDAC [78]. In [36], HUB4 1997 is employed, while in [40,78] MATBN2002 is used for testing. FAR in both [40,78] is deteriorated by 117.4% compared to that in [36]. Referring to MDR, it is improved in [40] by 19.0% with

respect to that in [36] and by 35.2% in [78] compared to that in [36]. The deterioration in FAR can be attributed to the different languages. HUB4 1997 contains English conversations, while MATBN2002 Mandarin ones. English and Mandarin exhibit significantly different tonal attributes. MDR improvement can be attributed to the fact that every change point has multiple chances to be detected in [40,78]. Finally, the second refinement step, that takes place after BIC segmentation, enhances the results in [78].

The real-time algorithm proposed in [36] is roughly comparable to the MDL algorithm [51]. In both cases the variable-size sliding analysis window strategy is applied. The first algorithm is tested on HUB4 1997 database, whereas the second one is evaluated on parts of the Topic Detection and TDT-3 Mandarin audio corpus. The problem of different languages is raised again. A relative MDR improvement of 59.5% in [51] compared to that reported in [36] is obtained at the expense of a 52.2% relative FAR deterioration. In [51], a smoothing procedure takes place aiming to reduce the number of false alarms. However, the FAR deterioration in [51] compared to [36] makes questionable the efficiency of the smoothing procedure.

With respect to the computationally efficient algorithms presented in [73] and the non-tuned algorithm of [41], they both utilize the heuristics introduced in [36] and extract MFCCs. F_1 appears to be relatively improved by 33.4% in [73] compared to that in [41]. This may be due to no tuning taking place in [41]. Concerning the computational cost, all the three methods in [73] aim to reduce the cost of estimating $\boldsymbol{\theta}_X$, $\boldsymbol{\theta}_Y$, and $\boldsymbol{\theta}_Z$. No such effort is made in [41].

An interesting outcome results when comparing the computationally efficient algorithms in [73] to the hybrid-algorithm proposed in [43]. The relative deterioration of F_1 in [43] when compared to [73] ranges from 49.4% for the "segment-level segmentation" case to 9.9% for the "segment-level segmentation, model-level segmentation, and HMM-based re-segmentation" case. However, one would expect a hybrid algorithm, such as that presented in [43], to yield better results than a metric-based algorithm [73], as claimed in [15,78].

Additionally, one could compare the sequential metric-based approach of [40] and the metric-SEQDAC algorithm in [78] to the MDL algorithm [51]. There is a relative FAR improvement in [51] equal to 30.0%, when compared to that in [40,78]. MDR in [51] is reduced to one half of that reported in [40]. This is consistent to the claim in [51] that MDL-based segmentation performs better than standard BIC-based segmentation. Moreover, several steps, like silence detection or smoothing the results, are undertaken in [51] to enhance the initial MDL-based results.

It is worth comparing the non-tuned algorithm proposed in [41] to the hybrid-algorithm demonstrated in [43]. PRC is relatively improved by 28.2%, RCL by 10.8%, and F_1 measure by 20.2% in [43] with respect to [41]. This could be attributed to the fact that the criterion proposed in [41] is not tuned. Improved

performance is expected also from a hybrid algorithm in general [15,78]. Concerning computational cost, the algorithm in [43] is of greater cost, since model-segmentation is also materialized.

2.5 Speaker segmentation outlook

2.5.1 Comments on BIC-based speaker segmentation algorithms

BIC-based speaker segmentation algorithms do not assume any prior knowledge of the number of the speakers, their identities, or signal characteristics. The only perquisite is that speakers do not speak simultaneously. BIC is applied to analysis windows of various durations exhibiting varying overlaps. The choice of the analysis window size N_Z is of great importance. On the one hand, if N_Z is too large, it may contain more than one speaker changes and consequently yield a high number of miss detections. On the other hand, if N_Z is too short, the lack of data will cause poor Gaussian estimation, especially of the covariance matrix [44], and, as a result, a poor segmentation accuracy. A typical initial analysis window duration is 2 s, which in most cases increases incrementally [35–37,39,41,42,44,51,53,73]. This is because researchers agree that BIC performance is poor, when two successive speaker changes are separated less than 2 s [7,8,36,37,44,51]. Based on this remark, several researchers adopt the continuous update of speakers models [7,8,35,40,42,78].

In BIC, as is defined in (16), $L_1 - L_0$ refers to the quality of the match between analysis windows X and Y, whereas the term $-\frac{\lambda}{2}\left(d + \frac{d(d+1)}{2}\right)\log N_Z$ is a penalty factor for model complexity. In coding theory, BIC with λ equal to 1 represents the shortest code length with which the data can be encoded. In speaker segmentation, λ serves as a threshold. Its choice is task-dependent and non-robust to different acoustic and environmental conditions. Consequently, λ requires tuning. Although the algorithm in [41] does not require any tuning, frequently it yields a poor accuracy. Concerning the way λ affects the figures of merit, the lower λ is, the larger PRC and MDR are and the lower FAR and RCL are. Long analysis windows yield high MDR and low FAR. Finally, a false alarm is more likely to occur in long homogenous segments than in short ones, when the same λ is used.

Concerning the figures of merit, the research community tends to treat false alarms as less cumbersome than missed detections. However, such a consideration highly depends on the application. Over-segmentation caused by a high number of false alarms, is easier to remedy than under-segmentation, caused by high number of miss detections [16,35,37,40,42,44,71,78]. Over-segmentation, for example, could be alleviated by clustering and/or merging. Equivalently, this means that PRC and FAR are easier to handle than RCL and MDR. This explains why λ is usually selected to yield a lower number of miss detections at the expense of a higher

number of false alarms.

Regarding computational cost, the full search implementation of BIC is computationally expensive, reaching $O(N_Z^2)$ [72]. The high computational cost motivates several researchers to employ heuristics. Heuristics are commonly applied to either compute $\boldsymbol{\theta}_X$, $\boldsymbol{\theta}_Y$, and $\boldsymbol{\theta}_Z$ fast [44,51,73] or conduct less BIC tests by selecting the time instants where the aforementioned tests are performed [36,37,41,44,73]. This is particulary true in real-time implementations [35,39,42,48].

In hybrid algorithms, BIC-based metric segmentation is applied prior to model-segmentation [43,78]. BIC is used as a segmentation criterion due to its high efficiency. The purer the initial segments, determined by BIC, the better results are expected from model-based segmentation.

2.5.2 Recommendations for speaker segmentation

Pre-segmentation improves BIC segmentation accuracy [8,37,40,44,78]. The main reason to perform pre-segmentation is to ensure that analysis windows are larger than 2 s, enabling BIC to yield more accurate results. An alternative for the small sample case is the BIC formulation proposed in [91]. Preliminary experiments indicate that BIC modifications on the model penalty factor lead to more accurate results for the small sample case. Posterior processing can also be used to further refine segmentation [15,43,51]. Such refinement techniques usually aim at lowering the number of false alarms.

An additional recommendation, is the use of additional diverse features. One should bear in mind that the Gaussianity assumption of the data samples $\mathbf{x}_i$ is not always correct. In [91], it is demonstrated that the generalized Gamma distribution fits the distribution of MFCCs better than the Gaussian one. Alternatively, BIC segmentation can be performed with various features and subsequent fusion of the results, since different features may complement in different contexts [7,35,42]. In addition, one could perform segmentation with various metrics and/or various classifiers, and then fuse the individual results. In general, fusion offers many advantages, such as increasing the reliability, robustness, and survivability of a system [86].

Heuristics reduce BIC computational cost. Most commonly used heuristics are: the varying analysis window scheme [36,41,44,51,73]; application of BIC tests not close to the borders of each analysis window [36,41,44,73]; omission of BIC tests at the beginning of large analysis windows [36,41,44,73]; frame skipping [36,41,44,51,73]; dynamic computation of the analysis window mean and covariance matrix [44,51,73]; and finally, the constant updating of the speaker model [7,8,35,42].

Finally, there is no obvious trend of the research community for using exclusively the set of figures of merit $\{PRC, RCL, F_1\}$ or the set $\{FAR, MDR\}$. Moreover,

there is no direct transformation from the one set to the other, unless the FA, MD, DET, and CFC are available. Thus, only reporting both sets of figures of merit facilitates comparisons [7,8].

3 Speaker Clustering

The discussion begins with the description of the evaluation measures employed to assess the performance of a clustering algorithm in subsection 3.1. Subsection 3.2 deals with the estimation of the number of clusters. Speaker clustering approaches are classified into two main categories: *deterministic* and *probabilistic* ones. The deterministic approaches cluster together similar audio segments with respect to a metric, whereas the probabilistic approaches use GMMs or HMMs to model the clusters. Subsection 3.3 is devoted to the description of deterministic speaker clustering algorithms, while subsection 3.4 focuses on probabilistic speaker clustering algorithms. A qualitative comparison of the reviewed speaker clustering algorithms is undertaken in subsection 3.5. Deductions and the authors insight to speaker clustering are discussed in subsection 3.6.

3.1 Evaluation measures for speaker clustering

Let

n_{ij} be the total number of audio segments in cluster i uttered by speaker j;
N_s be the total number of speakers;
N_c be the total number of clusters;
N be the total number of audio segments;
$n_{.j}$ be the total number of audio segments uttered by speaker j;
$n_{i.}$ be the total number of audio segments in cluster i.

The following equations establish relationships between the aforementioned variables:

$$n_{i.} = \sum_{j=1}^{N_s} n_{ij}, \quad n_{.j} = \sum_{i=1}^{N_c} n_{ij}, \quad N = \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} n_{ij}.$$

To evaluate the performance of a speaker clustering algorithm the following measures are used: the cluster purity and its average value; the cluster coverage; the speaker purity and its average value; the Rand index; the misclassification rate; the BBN metric; the overall diarization error measure, and the classification error.

Cluster purity

The purity of cluster i, $\pi_{i.}$ defined as [1,9,10,13,92]

$$\pi_{i.} = \sum_{j=1}^{N_s} n_{ij}^2 / n_{i.}^2 \tag{21}$$

and the average cluster purity given by [9,13,92]

$$acp = \frac{1}{N} \sum_{i=1}^{N_c} \pi_{i.}\ n_{i.} \tag{22}$$

provide a measure of how well a cluster is limited to only one speaker.

Cluster coverage

The cluster coverage is a measure of the dispersion of the data collected for a specific speaker across the clusters. It is defined as the percentage of the segments uttered by speaker j in cluster i, which has most of his data. It must be mentioned that cluster purity and cluster coverage are complementary measures [16,93].

Speaker purity

The speaker purity for speaker j, $\pi_{.j}$ [13]

$$\pi_{.j} = \sum_{i=1}^{N_c} n_{ij}^2 / n_{.j}^2 \tag{23}$$

and the average speaker purity for all speakers [13]

$$asp = \frac{1}{N} \sum_{j=1}^{N_s} \pi_{.j}\ n_{.j} \tag{24}$$

describe how well a speaker is limited to only one cluster.

Rand index

The Rand index assesses the consensus between two partitions. The first partition comes from a clustering solution, and the second is known a priori. The Rand index gives the *probability* that two randomly selected segments come from the same speaker but are hypothesized in different clusters or two segments are in the same

cluster but come from different speakers. It is defined as [10,94]

$$\gamma = \frac{1}{\binom{N}{2}} \left[\frac{1}{2} \sum_{i=1}^{N_c} n_{i.}^2 + \frac{1}{2} \sum_{j=1}^{N_s} n_{.j}^2 - \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} n_{ij}^2 \right] \tag{25}$$

with $\binom{N}{2}$ denoting the number of combinations of N segments by 2. It admits values between 0 and 1. A perfect clustering should yield a zero Rand index. However, the Rand index does not provide any information on how the partitions are distributed and how the partitions are related [94].

Misclassification rate

Given an one-to-one speaker-to-cluster mapping, if any segment from speaker j is not mapped to a cluster, an error is committed. Let e_j denote the total number of segments uttered by speaker j that are not mapped to the corresponding cluster. The misclassification rate (MR) defined as [10]

$$MR = \frac{1}{N} \sum_{j=1}^{N_s} e_j \tag{26}$$

ranges between 0 and 1. Small values of MR indicate a small probability of unmapped segments to any cluster.

BBN metric

The BBN metric is defined as [1]

$$I_{BBN} = \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} \frac{n_{ij}^2}{n_{i.}} - QN_c = \sum_{i=1}^{N_c} n_{i.}\ \pi_{i.} - QN_c = N\ acp - QN_c \tag{27}$$

where Q is a parameter specified by the user to quantify the preference on a few large clusters over the risk of merging clusters that do not really belong together. The larger the I_{BBN} is, the better performance is achieved.

Overall diarization error measure

A diarization system hypothesizes a set of speaker segments which are characterized by the corresponding start and end times and the related speaker-ID labels. The system is scored against the reference speaker segmentation, according to the ground truth information. This is performed by one-to-one mapping the reference speaker IDs to the hypothesized ones. *Missed speech* (MS) occurs when a speaker is present in reference but not in hypothesis, a *false alarm* (FA) occurs when a

speaker is present in hypothesis but not in reference, and finally, a *speaker error* (SE) occurs when the mapped reference speaker is not the same as the hypothesized one. The overall diarization error measure (DER) is defined as [17]

$$DER = MS + FA + SE. \tag{28}$$

Classification error

The classification error is defined as the percentage of time not attributed correctly to a reference speaker [95]. The error for cluster c_i, E_i, is defined as the percentage of the total time spoken by speaker i that has not been clustered to this cluster. The classification error, CE, is defined as [95]:

$$CE = \frac{1}{N_c} E_i \tag{29}$$

where N_c denotes the total number of clusters. CE admits values between 0 and 1. The smaller the CE value is, the better performance is achieved.

3.2 Automatic estimation of the number of the clusters

A difficult problem in speaker clustering is the estimation of the number of clusters to be created. Ideally, the number of clusters, N_c, should equal the number of speakers, N_s. However, N_c, derived by the algorithms, is generally greater than or equal to N_s. Two methods for automatically estimating the number of clusters are discussed next.

Voitovetsky et al. propose a validity criterion [96] for the automatical estimation of the number of speakers in Self Organizing Map (SOM)-based speaker clustering [97–99]. For efficient clustering, the intra-cluster distance should be relatively small, while the inter-cluster distance should be relatively large. Thus, the proposed criterion defines the validity of a given partition to be proportional to the ratio of the intra-cluster distance over the inter-cluster distance. Let n_m be the number of vectors in the mth segment. The distance between the ith and the pth SOMs for the nth vector, $Dcb_n(i,p)$, is defined as $Dcb_n(i,p) = [(\mathbf{c}_n(i) - \mathbf{c}_n(i,p))^T(\mathbf{c}_n(i) - \mathbf{c}_n(i,p))]^{1/2}$, where $\mathbf{c}_n(i)$ is the closest centroid of the ith SOM to the nth vector and $\mathbf{c}_n(i,p)$ is the closest centroid of the pth SOM to $\mathbf{c}_n(i)$. Let us also denote by $D_n(i)$ the Euclidean distance between the nth feature vector $\mathbf{x}_n$ and the closest centroid of the ith SOM defined as $D_n(i) = [(\mathbf{x}_n - \mathbf{c}_n(i))^T(\mathbf{x}_n - \mathbf{c}_n(i))]^{1/2}$. The contribution of the ith cluster to the validity coefficient is:

$$Q_i^{N_c} = \frac{1}{n_{i.}} \sum_{m \in i} \frac{1}{n_m} \sum_{n=1}^{n_m} \frac{D_n(i)}{\sum_{p=1,\ldots,N_c, p \neq i} n_{p.}\, Dcb_n(i,p)}. \tag{30}$$

The number of segments, m, can be determined by a speaker segmentation algorithm. The validity coefficient for N_c clusters, Q^{N_c}, is the sum of all contributions $Q_i^{N_c}$, i.e.

$$Q^{N_c} = \sum_{i=1}^{N_c} Q_i^{N_c}. \tag{31}$$

The estimated number of clusters minimizes the validity coefficient (31).

Another algorithm for the automatic estimation of the number of clusters in speaker clustering is based on the BIC [11]. Let $C_{N_c} = \{c_i : i = 1, \ldots, N_c\}$ be a clustering with N_c clusters and N be the number of the data samples to be clustered. Each cluster c_i is modeled by a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$ can be estimated by the sample mean vector and the sample dispersion matrix, respectively. Thus, the number of parameters for each cluster is $d + \frac{1}{2}d(d+1)$, with d denoting the data dimension as in (16). BIC for clustering C_{N_c} is defined as:

$$BIC(C_{N_c}) = \sum_{i=1}^{N_c} -\frac{1}{2} n_i. \log|\boldsymbol{\Sigma}_i| - N_c \, N\left(d + \frac{d(d+1)}{2}\right) \tag{32}$$

where $|\cdot|$ denotes the matrix determinant. The clustering which maximizes BIC is chosen. However, it is computational costly to search globally for the best BIC value. For hierarchical clustering methods, it is possible to optimize the BIC in a greedy fashion [11,16,17].

3.3 Deterministic methods for speaker clustering

Deterministic methods cluster together similar audio segments. The following subsection describes several well-known deterministic techniques.

3.3.1 SOM-based methods

SOMs are a powerful tool for speaker clustering. An algorithm for speaker clustering based on SOMs is proposed in [97–99]. The number of speakers N_s is assumed to be known. The data are divided into short segments. Each segment is considered to belong to only one speaker and to be long enough to enable determining speakers' identity. Segments of half a second are proven sufficient for good clustering. Several SOMs are used. A preliminary segmentation of the audio recordings into speech and non-speech segments is applied using thresholding. Non-speech segments are used to train a non-speech SOM. Furthermore, each of the N_s speakers is modeled by a Kohonen SOM of 6×10 neurons. Initially, speech segments are randomly and equally divided between the N_s models. The speech segments are clustered into N_s speakers by performing competition between the SOMs. Multiple

iterations are allowed during training. After each iteration, the data are re-grouped between the models. The training process is applied again to the new partition until the partitions remain unchanged or their difference between two consecutive iterations is less than a threshold value. At the end of the iterative procedure, the system yields $N_s + 1$ models. N_s models are devoted to the speakers and the last model to the non-speech data. Figure 1 depicts the general scheme of an unsupervised speaker clustering system, that yields $N_s + 1$ speaker models. However, the use of Figure 1 is not confined to SOM-based methods only.

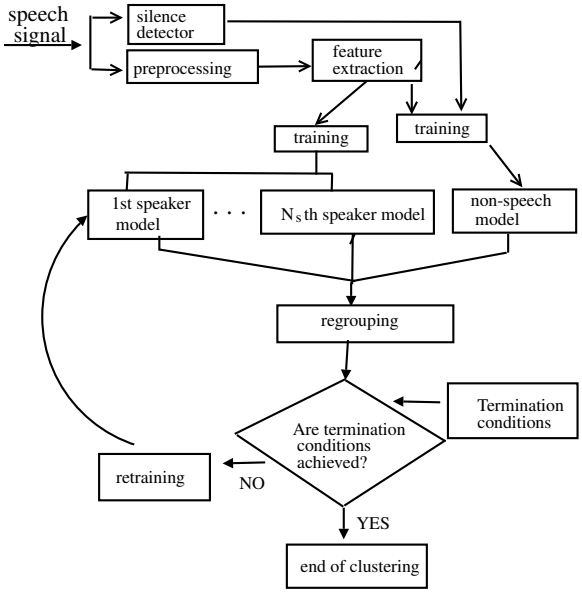

Fig. 1. General description of an unsupervised speaker clustering system.

The speech database used to test the algorithm is composed of clear speech and telephone Hebrew conversations [97–99]. Twelve LPC derived cepstrum coefficients and twelve delta-cepstrum coefficients are computed. Due to the finite resolution, some segments are split between speakers. As a result, a segment may contain data from two speakers. It has been observed that the classification error rate at splitting segments, equal to 5.9%, is higher than at non-splitting ones, equal to 2.8% for half second segments [97–99]. As the length of segments becomes shorter the appearance of splitting segments becomes rare. In general, the reported classification error for this system is 5.6% for two-speaker high-quality conversations and 6.2% for two-speaker telephone conversations [97–99].

3.3.2 Hierarchical methods

Liu and Kubala propose an on-line hierarchical speaker clustering algorithm [10]. Each segment is considered to belong exclusively to one speaker. The closest pairs of audio segments are found by comparing the distances among all the available segments. To calculate the distance between audio segments $\mathbf{s}_i$ and $\mathbf{s}_j$, the GLR is

used:

$$GLR(\mathbf{s}_i, \mathbf{s}_j) = \frac{L(\mathbf{s}_c; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{L(\mathbf{s}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) L(\mathbf{s}_j; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{33}$$

where $\mathbf{s}_c = \mathbf{s}_i \bigcup \mathbf{s}_j$; $\boldsymbol{\mu}_c$, $\boldsymbol{\mu}_i$, and $\boldsymbol{\mu}_j$ are the mean feature vectors in audio segments $\mathbf{s}_c$, $\mathbf{s}_i$, and $\mathbf{s}_j$, respectively; $\boldsymbol{\Sigma}_c$, $\boldsymbol{\Sigma}_i$, and $\boldsymbol{\Sigma}_j$ are the covariance matrices of the feature vectors; and $L(\cdot)$ is the likelihood of the data. Furthermore, the within-cluster dispersion $G(c)$ of cluster c is used, that is defined as

$$G(c) = |\sum_{i=1}^{N_c} n_i.\ \boldsymbol{\Sigma}_i| \sqrt{N_c}. \tag{34}$$

There are three variations of the on-line speaker clustering algorithm, namely the *leader-follower clustering (LFC)*, the *dispersion-based speaker clustering (DSC)*, and the *hybrid speaker clustering (HSC)*. The LFC algorithm alters only the cluster centroid which is the most similar to the new pattern being presented or creates a new cluster if none of the existing clusters is similar enough to it. LFC uses GLR (33) and a threshold ϑ in order to decide if two segments are close. The DSC algorithm does not use any threshold, but it resorts to the within-cluster dispersion $G(c)$ (34). Even though DSC has a tendency to underestimate the number of clusters and consequently does not perform well, threshold independence makes it very appealing. Furthermore, the DSC method assumes that there is not any interference between speakers [10]. The HSC algorithm uses both within-cluster dispersion $G(c)$ and GLR. The data used for experiments are from HUB4 1998 corpus [105]. The evaluation metrics used to assess the algorithm performance are cluster purity (21), Rand index (25), and misclassification rate (26). The thresholds for LFC and HSC are tuned on HUB4 1996 test set [104] by minimizing the overall misclassification rate. The best values measured for the aforementioned metrics using LFC are 0.02, 1, and 0.01, respectively; 0.06, 0.986, and 0.031, respectively for DSC; and finally 0.06, 0.999, and 0.011, respectively for HSC, as can be seen in Table 7 [10]. Experiments demonstrated that the DSC algorithm performs worst with respect to all measures, suggesting that within-cluster dispersion alone might not be a good criterion for on-line speaker clustering. Concerning the off-line hierarchical clustering algorithms, they require all the data to be available before clustering. Intuitively, off-line speaker clustering should work better than on-line, because it exploits more information.

A deterministic step-by-step speaker diarization system is developed by Meignier et al. [15]. It is based on speaker segmentation followed by hierarchical clustering. The number of speakers is automatically estimated. The first step of the algorithm is the macro-class acoustic segmentation, that divides the audio into four acoustic classes according to different conditions based on gender and wide-/narrow-band detection. Furthermore, silence and non-speech segments are removed. The system

consists of three modules. The first module performs speaker segmentation using a distance metric approach based on (33). A GLR distance curve is computed with non-overlapping windows 1.75 s long. The maximum peaks of the curve are the most likely speaker change points. The created segments are input to the hierarchical clustering module. Initially, a UBM is trained on the available data. Afterwards, segment models are trained using maximum a posteriori (MAP) adaptation of the UBM. GLR distances are then computed between models and the closest segments are merged until N_s segments are left [15]. Clustering is performed individually on each acoustic macro-class and the results are finally merged. The third module applies the penalized BIC criterion in order to estimate the number of speakers [15]. The data used for experiments come from HUB4 1998 corpus by discarding the advertisements (ELISA-Dev) and from channels PRI (Public Radio International), VOA (Voice of America), and MNB (MSNBC), also discarding advertisements (ELISA-Eva) [15]. 16 MFCCs computed every 10 ms with 20 ms windows using 56 filter banks, and the energy are extracted from the speech signal. The evaluation measure used to assess the algorithm performance is DER (28). The best DER equals 10.2%. It is obtained on ELISA-Eva data set, when the acoustic macro-class segmentation is performed manually [15].

3.4 Probabilistic methods for speaker clustering

Probabilistic methods use GMMs or HMMs to build models that describe the clusters.

3.4.1 GMM-based methods

Many approaches based on GMMs have been proposed. Tsai et al. propose a speaker clustering method which is based on the voice characteristic reference space [9]. The reference space aims at representing some generic characteristics of speaker voices derived through training. The speech features are projected onto a reference space so that they are clustered. The projection vectors reflect the relationships between all segments. They are more robust against the interference from non-speaker factors. Three distinct methods are proposed for the construction of the projection vectors [9]: the *utterance-individual Gaussian mixture modeling*, the *utterance-universal vector clustering*, and the *utterance-universal Gaussian mixture modeling followed by utterance-individual model adaptation*.

The general mathematical framework is described next. Let $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$ be N unlabelled speech segments. Each segment is uttered by one of N_s speakers. The reference space can be created either by using the N segments to be clustered or by another arbitrary speech data set and is composed of K bases [9]. Each basis refers to representative voice characteristics encoded by spectrum-based features.

After having constructed the reference space, each segment $\mathbf{s}_i$ is mapped onto a K dimensional projection vector $\mathbf{V}_i = [v(\mathbf{s}_i, \boldsymbol{\phi}_1), v(\mathbf{s}_i, \boldsymbol{\phi}_2), \ldots v(\mathbf{s}_i, \boldsymbol{\phi}_K)]^T$, where $v(\mathbf{s}_i, \boldsymbol{\phi}_k)$ denotes how much segment $\mathbf{s}_i$ can be characterized by the basis vector $\boldsymbol{\phi}_k$. If two segments $\mathbf{s}_i$ and $\mathbf{s}_j$ stem from the same speaker, the majority of the projection values in $\mathbf{V}_i$ and $\mathbf{V}_j$ will be similar. The similarity between any two segments $\mathbf{s}_i$ and $\mathbf{s}_j$ is computed using the cosine similarity measure

$$CSM(\mathbf{s}_i, \mathbf{s}_j) = \frac{\mathbf{V}_i^T \mathbf{V}_j}{\|\mathbf{V}_i\| \, \|\mathbf{V}_j\|}. \tag{35}$$

The segments which are similar enough are grouped into a cluster.

The utterance-individual Gaussian mixture modeling, uses one GMM for each of the N segments to be clustered. The resulting N GMMs, $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N$, form reference bases $\boldsymbol{\phi}_k = \boldsymbol{\theta}_k, k = 1, 2, \ldots, N$. For each segment $\mathbf{s}_i$, its projection to basis $\boldsymbol{\phi}_k$ is computed by $v(\mathbf{s}_i, \boldsymbol{\phi}_k) = \log p(\mathbf{s}_i|\boldsymbol{\theta}_k)$. The utterance-universal vector clustering, uses a single utterance-independent codebook having K codewords. The codebook can be considered as a universal model trained to cover the speaker-independent distribution of feature vectors. Each codeword $\mathbf{cw}_k, k = 1, 2, \ldots, K$ consists of a mean vector $\boldsymbol{\mu}_k$ and a diagonal covariance matrix $\boldsymbol{\Sigma}_k$. Codebook training is performed via K-means using the Mahalanobis distance. After having created K codewords, each feature vector is explicitly assigned a codeword index. The projection value $v(\mathbf{s}_i, \boldsymbol{\phi}_k)$ is computed as

$$v(\mathbf{s}_i, \boldsymbol{\phi}_k) = \frac{\#\text{ feature vectors in } \mathbf{s}_i \text{ assigned to } \boldsymbol{cw}_k}{\#\text{ feature vectors in } \mathbf{s}_i} \tag{36}$$

where $\#$ denotes set cardinality.

The utterance-universal Gaussian mixture modeling followed by utterance-individual model adaptation, creates an universal GMM using all the segments to be clustered, followed by an adaptation of the utterance-universal GMM performed for each of the segments using the MAP estimation.

The speech data used in experiments consist of 197 speech segments chosen from 2001 NIST Speaker Recognition Evaluation Corpus [102]. The speech features include 24 MFCCs extracted from 20 ms Hamming windowed frames with 10 ms frame shifts. The measures used to evaluate the performance of the algorithm are cluster purity (21), average cluster purity (22), and Rand index (25). When the number of clusters is equal to the speaker population ($N_c = N_s = 15$), the best acp and Rand index γ achieved are 0.69 [9] and 0.0674, respectively, by the utterance-universal Gaussian mixture modeling followed by utterance-individual model adaptation method (ADA). The best acp and Rand index values achieved by the utterance-individual Gaussian mixture modeling method are 0.67 and 0.09, respectively, and by the utterance-universal vector clustering method 0.5 and 0.09, respectively [9].

Solomonoff et al. also propose a method for clustering speakers based on GMMs [1]. Each speaker is modeled by a GMM. The models are trained using the EM algorithm to refine the weight and the parameters of each component. The algorithm has three stages. The first stage aims at computing some distance-like measure between each pair of speech segments, such as the GLR (33) and the cross entropy defined as [1]:

$$d_{CE}(\mathbf{s}_0, \mathbf{s}_1) = \log \frac{L(\mathbf{s}_0|\boldsymbol{\theta}_{\mathbf{s}_0})}{L(\mathbf{s}_0|\boldsymbol{\theta}_{\mathbf{s}_1})} + \log \frac{L(\mathbf{s}_1|\boldsymbol{\theta}_{\mathbf{s}_1})}{L(\mathbf{s}_1|\boldsymbol{\theta}_{\mathbf{s}_0})} \tag{37}$$

where $\boldsymbol{\theta}_{\mathbf{s}}$ denotes the model trained (using EM) on segment $\mathbf{s}$ and $L(\mathbf{s}|\boldsymbol{\theta}_{\mathbf{s}})$ denotes the likelihood of the segment $\mathbf{s}$ with respect to the model $\boldsymbol{\theta}_{\mathbf{s}}$.

A tree or dendrogram of clusters is created at the second stage, where each segment forms its own cluster at the beginning, and clusters are merged recursively according to (33) or (37). The last stage picks one of the partitions, a process is called *dendrogram cutting*. The quality of the partition is measured by the BBN metric (27). The database used is SWITCHBOARD corpus [88]. The utterance set contains utterances from 20 speakers. The feature vectors used in the experiments have 38 elements, 19 LPC derived cepstrum coefficients and the corresponding delta coefficients. This method assumes that simultaneous speech does not occur. The performance of the method depends on dendrogram cutting. The best BBN metric value measured is 42 for a partition with 29 clusters setting $Q = 1/2$ [1].

A clustering method that is based on maximum purity estimation, which aims to maximize the total number of within-cluster segments from the same speakers, is proposed in [92]. Maximum purity estimation is motivated by the fact that although hierarchical clustering guarantees the homogeneity of individual clusters, it is not guaranteed for all clusters. The method employs a genetic algorithm to determine the cluster where each segment should be assigned to. First, the optimal number of clusters is estimated using BIC, as was explained in Section 3.2. Second, the inter-utterance similarities are calculated and, finally, the segments that are similar enough to be grouped into a cluster are determined. Initially, a GMM which represents the generic characteristics of all speakers' voices is created using the cepstral features of the N segments to be clustered. This GMM is adapted so that it models the individual voice characteristics using MAP estimation. Therefore, N utterance-dependent GMMs are created. Next, N super-vectors are constructed by concatenating all the mean vectors of the utterance-dependent GMMs in the order of the mixture index. Afterwards, principal component analysis is applied on the super-vectors yielding E eigenvectors, that create a voice characteristic space. Each segment is described by its coordinates $\mathbf{q}_i$ on the voice characteristic space. The similarity between segments $\mathbf{s}_i$ and $\mathbf{s}_j$ is calculated using the cosine similarity measure (35) between $\mathbf{q}_i$ and $\mathbf{q}_j$. Next, a genetic algorithm is applied in order to derive the clustering with the maximum cluster purity. Speech data from 2001 NIST Speaker Recognition Evaluation Corpus are used [102]. The features include 24

MFCCs extracted from audio frames of duration 20 ms that overlap by 10 ms. The Hamming window is used to extract the audio frames. The evaluation measure is the average cluster purity (22). When the number of clusters is equal to the speaker population ($N_s = N_c = 15$), the best acp measured is 0.81 [92].

Jin et al. propose a fully automated speaker clustering algorithm [103]. Let us consider a collection of segments $\mathbf{s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$. Cepstral feature vectors are extracted from each segment. The algorithm assumes that the vectors in each of these segments are statistically independent and can be modeled by a multivariate Gaussian distribution. A distance matrix is built based on Gaussian models of acoustic segments. The distance measure introduced by Gish et al. [79] is used. Next, hierarchical clustering is performed to generate a list of clustering solutions for combinations of α and N_c, where α is a scaling parameter, which corresponds to the likelihood consecutive segments come from the same speaker. The hierarchical clustering procedure takes the distance matrix as input and continues to aggregate clusters together until one large enough cluster is formed. The output is a tree of clusters, which can be pruned for any given number N_c. Finally, model selection is conducted by employing a criterion that penalizes too many clusters. The penalized criterion for N_c clusters is $B = \sqrt{N_c}\ G(c)$, where $G(c)$ denotes the within-cluster dispersion (34). An efficient clustering should have a relatively small dispersion within clusters. The database used in experiments is HUB4 1996 [104]. Each speech segment is chopped into shorter ones, so that each segment contains 20 words on average. The algorithm performance is evaluated with respect to the word error rate (WER). The WER, as its name implies, measures the number of words that differ between the hypothesis and the reference. The smallest WER measured is 24.8% on chopped audio segments [103]. However, the algorithm tends to underestimate the number of clusters.

Lu and Zhang propose an unsupervised speaker segmentation and tracking algorithm in real-time audio content analysis [42]. No prior knowledge of the number and the identities of speakers is assumed. The algorithm first performs speaker segmentation to find speaker change points, and then speaker tracking, which clusters speech segments with respect to speaker identities. It is composed of four modules: the front-end processing module, the segmentation module, the clustering and speaker model updating module, and the speaker tracking module. The input audio stream is assumed to be speech only. In the front-end process, the input speech stream is divided into 3 s segments with 2.5 s overlapping. LSPs, MFCCs, and pitch are extracted from the speech segments. These features are then fused in a parallel Bayesian network. Afterwards, speaker segmentation is performed in a "coarse to refine" manner, where a potential speaker change point is first detected and then validated. An initial Gaussian model is estimated for each segment, and then the divergence between every two consecutive zero-mean Gaussian models is calculated, i.e.

$$D(\mathbf{\Sigma}_i, \mathbf{\Sigma}_j) = \frac{1}{2}\mathrm{tr}[(\mathbf{\Sigma}_i - \mathbf{\Sigma}_j)(\mathbf{\Sigma}_j^{-1} - \mathbf{\Sigma}_i^{-1})] \tag{38}$$

where $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_j$ are the estimated covariance matrices of the ith and jth segments, respectively, and tr$[\cdot]$ denotes the trace of a matrix. A potential speaker change point between two consecutive speech segments $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$ is detected, if there exists a local peak in (38) and $D(\mathbf{\Sigma}_i, \mathbf{\Sigma}_{i+1})$ exceeds a threshold. The threshold is dynamic and data-dependent [42]. When no potential speaker change point is identified, the data of the current segment are assigned to the current speaker in order to yield a more accurate speaker model. If a potential speaker change boundary is detected, Bayesian fusion is used to confirm if it is really a speaker change boundary [42]. Finally, when a speaker change boundary is validated, the algorithm searches the speaker models created thus far to identify the newly appeared speaker. If the speaker can not be identified, a new speaker model is created. Otherwise, the identified speaker model is updated with the new speaker data. Let $\mathcal{M}$ be the model of the speaker to be identified. To identify the speaker, the current segment is compared with all existing speaker models to find which model is the most similar to the current segment. The dissimilarity between an existing speaker model and the current segment is set as the weighted sum of the $\mathcal{K}$ nearest Gaussian components to the speaker model, i.e. $D' = \sum_{i \in J(\mathcal{K})} \nu_i D(\mathbf{\Sigma}_i, \mathbf{\Sigma}_{\mathcal{M}})$, where ν_i is the weight of the ith component in $J(\mathcal{K})$, the set of $\mathcal{K}$ nearest Gaussian components to $\mathbf{\Sigma}_{\mathcal{M}}$, $J(\mathcal{K}) = \{i | D(\mathbf{\Sigma}_i, \mathbf{\Sigma}_M) < D_{(\mathcal{K})}\}$, and $D_{(\mathcal{K})}$ is the $\mathcal{K}$th smallest distance in the series $D(\mathbf{\Sigma}_i, \mathbf{\Sigma}_{\mathcal{M}}), i = 1, 2, \ldots, l$, assuming that l components are found in the ith speaker model. The flow diagram of the algorithm is illustrated in Figure 2. HUB4 1997 is used for experiments [106]. The algorithm can recall 89% of speaker change points with 15% false alarms and 76% of speakers can be unsupervisedly identified with 20% false alarms [42]. The results shown in Table 7 under the label F-R ($\%$), correspond to the best reported speaker change false alarm rate and the corresponding speaker change recall rate, defined in (17) and (18), respectively [42].

An iterative segmentation/clustering procedure (c-std) is proposed in [16,93,107]. Each initial segment is used to seed one cluster and an eight-component GMM with diagonal covariance matrices is trained using the speech segments. Then, given a sequence of N segments $\mathbf{s}_i, i = 1, 2, \ldots, N$ and the corresponding cluster labels $l_i \in [1, N_c]$ with $N_c \leq N$ and ideally $N_c = N_s$, the following objective function is maximized:

$$\sum_{i=1}^{N} \log L(\mathbf{s}_i | \boldsymbol{\theta}_{c_i}) - \alpha \; N - \beta \; N_c \tag{39}$$

where $L(\mathbf{s}_i | \boldsymbol{\theta}_{c_i})$ is the likelihood of the segment $\mathbf{s}_i$ given the model of its cluster c_i, and α and β are segment-related and cluster-related penalties. The algorithm stops when no more merges between GMMs can be done. It is tested on data combined from HUB4 1996 [104] and HUB4 1997 [106]. 38 features are extracted from the speech signal every 10 ms using a 30 ms frame on a 8 kHz frequency band, namely 12 MFCCs, 12 delta MFCCs, 12 delta-delta MFCCs, plus the delta and delta-delta log-energy. The algorithm performance is evaluated with respect to the average cluster purity (22), cluster coverage, and DER (28). When $\alpha = \beta = 230$, the acp

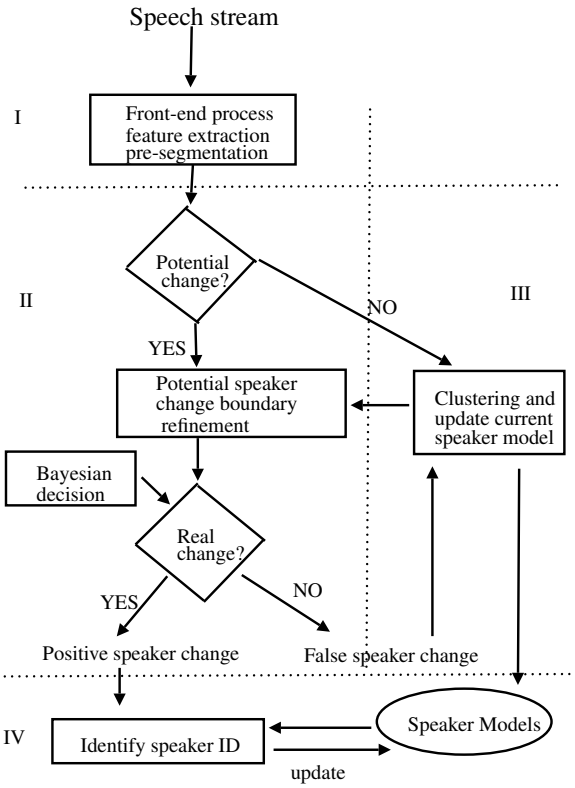

Fig. 2. The flow diagram of the algorithm, composed of four modules: I. Front-end process. II. Speaker segmentation. III. Clustering and speaker model updating. IV. Speaker tracking (adapted from [42]).

measured is 0.906, the cluster coverage is 82.1%, and DER is 24.8% [16,93,107].

In order to improve the clustering performance of the iterative GMM segmentation/clustering procedure, a BIC-based clustering algorithm (c-bic) is tested in [16,107]. At the beginning, each segment becomes the seed of one cluster, modeled by a Gaussian component with a full covariance matrix, and in the following steps, the clusters are merged until a stop criterion is reached. Two clusters c_i and c_j are merged, when the ΔBIC value (40) is a negative number:

$$\Delta BIC = (n_{i.} + n_{.j}) \log |\mathbf{\Sigma}| - n_{i.} \ \log |\mathbf{\Sigma}_i| - n_{.j} \ \log |\mathbf{\Sigma}_j| - \lambda \ P_B \qquad (40)$$

where λ is a data-dependent penalty factor as in (16), $\mathbf{\Sigma}$ is the covariance matrix of the merged cluster, $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_j$ are the covariance matrices of clusters c_i and c_j, respectively, and $n_{i.}$ and $n_{.j}$ are the number of frames in clusters c_i and c_j. P_B denotes the penalty factor (16). The clustering procedure stops when no more merges between clusters can be done. The algorithm is tested on data combined from HUB4 1996 [104] and HUB4 1997 [106]. Static and delta coefficients are extracted from the speech signal. When $\lambda = 5.5$, the acp measured is 0.971, the cluster coverage is 90.2%, and DER is 13.2% [16,93,107].

Recently, cluster recombination (c-sid) is proposed in [17,107]. It ends up with less clusters than the reference ones, but still the created clusters contain reasonable amounts of speech data. A UBM is built on the training data to represent general speakers. Each cluster from the UBM is updated using MAP adaptation, in order to build a single model per cluster. The cross likelihood ratio (CLR) between any two given clusters is computed as [107]

$$CLR(c_i, c_j) = \log\left(\frac{L(\mathbf{s}_i|\boldsymbol{\theta}_j)\ L(\mathbf{s}_j|\boldsymbol{\theta}_i)}{L(\mathbf{s}_i|\boldsymbol{\theta}_{ubm})\ L(\mathbf{s}_j|\boldsymbol{\theta}_{ubm})}\right) \tag{41}$$

where $L(\mathbf{s}_i|\boldsymbol{\theta}_j)$ is the average likelihood per frame of segment $\mathbf{s}_i$ given the model $\boldsymbol{\theta}_j$ (37). The pair of clusters with the highest CLR is merged to create a new model. The process stops when the CLR between all clusters is below a predefined data-dependent threshold thr. The data used to assess the algorithm are from HUB4 1996 [104] and HUB4 1997 [106]. The feature vectors consist of energy, 14 MFCCs plus delta MFCCs, and delta energy. Afterwards, feature normalization is performed in each segment. The algorithm performance is evaluated with respect to the average cluster purity, cluster coverage, and DER. When $\lambda = 3.5$ and $thr = 0.1$, the acp measured is 0.979, cluster coverage is 95.8%, and DER is 7.1% [16,93,107].

3.4.2 HMM-based methods

HMMs have been widely used in speaker clustering. Ajmera et al. propose an HMM-based speaker clustering algorithm [13]. Each state of the HMM represents a cluster and the pdf of each cluster is modeled by a GMM. The HMM is trained using the EM algorithm. The initialization of the pdfs is done using the K-means algorithm. The technique starts with over-clustering the data in order to reduce the probability that different speakers are clustered into one class. Afterwards, the segmentation is performed using the Viterbi algorithm in each cluster. The next step is to reduce the number of clusters by merging. The clusters are merged according to a likelihood ratio distance measure, such as (33) and (37). The new class is represented by another GMM having a number of components equal to the sum of the components of the individual clusters. The parameters of this newly formed cluster are retrained by the EM algorithm using the features belonging to the clusters to be merged. The segmentation is re-estimated with the new HMM topology having one cluster less and the likelihood of the data based on this segmentation is calculated. The likelihood increases, if the data in the two clusters to be merged are from the same speaker. On the opposite, it decreases if the clusters to be merged have data from different speakers. The merging process stops when the likelihood does not decrease any more.

This method uses only highly voiced audio segments. An audio segment is identified as voiced or unvoiced using the auto-correlation. The number of speakers and the speaker change points are assumed to be known a priori. The measures used for the evaluation performance of the algorithm are cluster purity (21), average cluster purity (22), speaker purity (23), average speaker purity (24), and an overall evaluation criterion A defined as $A = \sqrt{acp\ asp}$. The presence of non-speech produces many extra clusters, especially when non-speech comes from music, noise, or clapping. The proposed clustering algorithm is tested on HUB4 1996 evaluation data [104]. The speech features are LPCs. The clustering performance depends on the initial over-clustering. The algorithm must start with a sufficiently large number of clusters. The use of highly voiced segments only results in reduced computational complexity. Using all segments, the best acp and asp values reported equal 0.85 and 0.83, respectively. On average, asp and acp are greater than 0.7 [13].

Ajmera and Wooters propose another robust speaker clustering algorithm [95]. The algorithm automatically performs both speaker segmentation and clustering without any prior knowledge of the speaker identities or the numbers of speakers. The algorithm uses HMMs, agglomerative clustering, and BIC. The algorithm does not require any threshold and, accordingly, training data. Figure 3 shows the HMM topology used for clustering. The number of states in the HMM is equal to the initial number of clusters. Each state is composed of a set of S sub-states. The sub-states impose a minimum duration on the model. Each state of the HMM is a cluster and is expected to represent a single speaker. The pdf of each state is a GMM with M_m Gaussian components, which are shared among all sub-states. The algorithm starts with over-clustering the data into N_c clusters with $N_c > N_s$. The first step is to initialize the parameters of HMM. The initialization is performed using a uniform segmentation of the data in terms of the N_c clusters and estimating the parameters of the cluster GMMs over these segments. Furthermore, the K-means algorithm can be used for initialization. The next step is to train the HMMs using the EM algorithm. In the E-step, a segmentation of the data is obtained to maximize the likelihood of the data, given the parameters of the GMM. In the M-step, the parameters of the GMM are re-estimated based on the new segmentation. The final step is cluster merging. The optimal number of clusters N_c must be found. Ideally, N_c should equal N_s. BIC can be used as a merging criterion. Alternatively, a new merging criterion, merges a pair of clusters (c_1, c_2) if the following inequality is satisfied [95]:

$$\log p(c|\boldsymbol{\theta}) \geq \log p(c_1|\boldsymbol{\theta}_1) + \log p(c_2|\boldsymbol{\theta}_2) \tag{42}$$

where c_1 and c_2 represent the data in two clusters, $c = c_1 \bigcup c_2$ are the total data in the clusters, $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ represent the parameters of the pdfs of the two clusters, and $\boldsymbol{\theta}$ are the parameters of the pdf of c.

The algorithm is tested on three different data sets released by NIST, namely data used for preliminary experiments (dryrun) [108], data used as development data (devdata) [108], and data used in the final evaluation (evaldata) [108]. The dryrun data consist of six 10-minute excerpts from English broadcast news, while the dev-

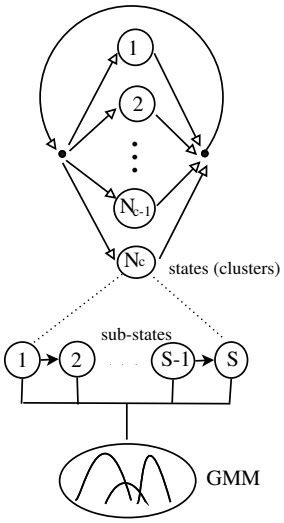

Fig. 3. HMM topology used for clustering in [95].

data and evaldata consist of three half-hour English broadcast news audio segments each [108]. 19 MFCCs are extracted, because they proved to work better than LPCs in noisy environments. The experimental results reveal that the algorithm is not sensitive to the minimum duration, the type of initialization, and the number of the Gaussians per initial cluster M_m. A rule of thumb, that was experimentally found, is to choose the number of clusters N_c equal to or greater than the duration of the audio data for English broadcast news in minutes. The evaluation measure used is the classification error. The best error value measured is 20.79% on evaldata [95].

Meignier et al. [15] also propose an integrated speaker diarization system, that generates an HMM which detects and adds a new speaker. The speaker detection is performed in four steps. In the first step, a one-state HMM $\mathcal{M}_0$ is initialized and used to model all speakers. In the second step, a speaker is extracted from $\mathcal{M}_0$ and the new speaker model $\mathcal{M}_x$ is trained using the 3 s region from $\mathcal{M}_0$ that maximizes the likelihood ratio between $\mathcal{M}_0$ and a UBM. The selected 3 s are moved from $\mathcal{M}_0$ to $\mathcal{M}_x$ in the segmentation hypothesis. In the third step, the speaker models are adapted according to the current segmentation and afterwards, the Viterbi decoding produces a new segmentation. Adaptation and decoding are iterated while the segmentation differs between two successive adaptation/decoding steps. Two segmentations are different, whenever at least one frame is assigned to two different speakers [15]. In the final step, the speaker model is validated. The likelihood of the previous solution and the likelihood of the current solution are computed and compared [15]. The stopping criterion is reached when no gain in terms of likelihood is observed or when no more speech is left to initialize a new speaker. In order to minimize DER some heuristics can be applied [15]. The data used for the experiments are from HUB4 1998 corpus without the advertisements (ELISA-Dev) and from channels PRI, VOA, and MNB, where advertisements are also discarded (ELISA-Eva) [15]. 20 MFCCs, computed every 10 ms on 20 ms windows using 56 filter banks, and the normalized energy are extracted from the speech signal. The

evaluation measure used to assess the algorithm performance is DER (28). The best DER, equal to 10.7%, is obtained on ELISA-Dev data set, when the acoustic macro-class segmentation is performed manually [15].

3.4.3 Algorithm proposed by the authors

A speaker diarization system is proposed by the authors in [109]. The system is composed of four serially connected modules: the speaker segmentation module, the speaker modeling module, the clustering module, and the cluster merging module. The system makes no assumptions on the number of speakers participating in the conversation. Non-speech and silent frames have been automatically filtered out. 24-order MFCCs are employed. In the first module, a BIC-based speaker segmentation algorithm is applied [7]. Over-segmentation is strongly preferred against the risk of not detecting true speaker change points. The resulted speech segments constitute the input of the speaker modeling module. The feature vectors within each speech segment are treated as i.i.d. random vectors, thus the speech segments can be modeled by multivariate Gaussian distributions. The clustering module, utilizes cluster ensembles in order to reveal the natural number of clusters. Three hierarchical algorithms are utilized to produce the cluster ensemble, namely the average group linkage, the weighted average group linkage and the Ward's hierarchical clustering method. The distance between the speech segments is calculated and the co-association matrix is computed. Each speech segment is considered to belong to the same cluster with another segment, when the respective entry of the co-association matrix exceeds a threshold. Two variants of the algorithm are available. The first variant does not cluster the speech segments that are considered to be outliers, while the second one does. The final module, merges together the most similar clusters. The data used for experiments consist of 10 movie dialogue scenes extracted from 5 movies, namely *Analyze That, Cold Mountain, Jackie Brown, Lord of the Rings I*, and *Secret Window* with a total duration of 10 min and 19 s. The algorithm is evaluated with respect to the classification error. The best mean classification error value is 14.783%, yielding the creation of 7 clusters, when outliers are not included in the clustering. Obviously, the two natural clusters are split into more than one sub-clusters. Generally, the algorithm tends to overestimate the number of clusters. Our system outperforms the system presented in [95], which employs the same features and evaluation criterion, but on different data set.

3.5 Comparative assessment of speaker clustering algorithms

A qualitative comparison of the described approaches for speaker clustering is undertaken next. Table 7 summarizes the data, the features, and the evaluation criteria used by each algorithm. As it can be noticed, each algorithm utilizes different data, different features, and different evaluation measures for performance assessment.

Though a strict assessment is not possible, some performance indications still can be deduced.

Table 7
Comparative results of the speaker clustering algorithms.

Method	Data	Features	Performance Criteria									
			acp	asp	Rand index	MR	F-R (%)	WER (%)	cluster coverage (%)	DER (%)	BBN metric	CE (%)
Voitovetsky et al. [97] and Lapidot et al. [98,99]	recorded Hebrew conversations [98,99]	12 first LPCs and their delta coefficients	-	-	-	-	-	-	-	-	-	5.60
Liu et al. (LFC) [10]	HUB4 1998 [105]	not specified	1.000	-	0.001	0.02	-	-	-	-	-	-
Liu et al. (DSC) [10]			0.986	-	0.031	0.06	-	-	-	-	-	-
Liu et al. (HSC) [10]			0.999	-	0.011	0.06	-	-	-	-	-	-
Meignier et al. (step-by-step system) [15]	ELISA-Eva [15]	16 MFCCs and energy	-	-	-	-	-	-	-	10.2	-	-
Tsai et al. (GMM) [9]	2001 NIST [102]	24 MFCCs	0.500	-	0.090	-	-	-	-	-	-	-
Tsai et al. (VC) [9]			0.670	-	0.090	-	-	-	-	-	-	-
Tsai et al. (ADA) [9]			0.690	-	0.067	-	-	-	-	-	-	-
Solomonoff et al. [1]	SWITCHBOARD [88]	19 Cepstral and their delta coefficients	-	-	-	-	-	-	-	-	46	-
Tsai and Wang [92]	2001 NIST [102]	24 MFCCs	0.810	-	-	-	-	-	-	-	-	-
Jin et al. [103]	HUB4 1996 [104]	Cepstral coefficients	-	-	-	-	-	24.8	-	-	-	-
Lu and Zhang [42]	HUB4 1997 [106]	MFCCs, LSPs, pitch	-	-	-	-	15-89	-	-	-	-	-
Barras et al. (c-std) [16,93,107]	combined data from HUB4 1996 [104] and HUB4 1997 [106]	12 Cepstral and their delta coefficients, delta and delta-delta energy	0.906	-	-	-	-	-	82.1	24.8	-	-
Barras et al. (c-bic) [16,93,107]		not specified	0.971	-	-	-	-	-	90.2	13.2	-	-
Barras et al. (c-sid) [16,93,107]		15 MFCCs and their delta coefficients, delta energy	0.979	-	-	-	-	-	95.8	7.10	-	-
Ajmera et al. [13]	HUB4 1996 [104]	LPCs	0.850	0.830	-	-	-	-	-	-	-	-
Ajmera and Wooters [95]	devdata [108] evaldata [108]	19 MFCCs	-	-	-	-	-	-	-	-	-	20.79
Meignier et al. (integrated system) [15]	ELISA-Dev [15]	20 MFCCs and normalized energy	-	-	-	-	-	-	-	10.7	-	-

First of all, it should be noted that the efficiency of a speaker clustering algorithm is greatly affected by the way the segments to be clustered have been created. That is, whether the segmentation of the input speech stream has been performed manually or automatically. The former case, even though does not introduce segmentation errors, does not possess a clear practical value due to human intervention. In the latter case, automatic segmentation yields errors that can degrade clustering per-

formance. Moreover, the minimum segment duration can play an important role in speaker clustering performance, since the longer the segment is, the more speaker information is available. However, there is always the risk that long segments might contain data from more than one speaker. Non-speech removal is another crucial step for speaker clustering, since non-speech segments can lead to many additional clusters. Finally, algorithms that assume an a priori known number of speakers yield an improved speaker clustering performance, although such a case is seldom met in practical applications.

SOM-based methods [97–99] split the speech stream into 0.5 s segments containing speech from more speakers. In addition, speech/non-speech discrimination is essential, as described in subsection 3.3. The low classification error achieved can be explained by the fact that these algorithms exploit the a priori known number of clusters. The hierarchical algorithm presented in [10] assumes that the segmentation step has been performed manually, before clustering. This might explain the high performance, while the algorithm in [15] performs speaker segmentation before the clustering step. The hierarchical algorithms automatically estimate the number of clusters in contrast with SOM-based ones [10].

Most of the GMM-based methods [1,9,92,103] cluster together segments that have emerged from a previous segmentation step, while segmentation and clustering are performed in the same time, using 3s segments in [42]. In addition, all algorithms automatically determine the number of speakers. The method in [103] underestimates the number of clusters, a fact that might justifies the high WER. The methods in [1,9,42] require non-speech removal, that can degrade clustering performance.

All HMM-based methods [13,15,16,93,95,107] automatically estimate the number of clusters and assume (or perform) segmentation preceding clustering. Furthermore, all algorithms perform speech/non-speech detection.

Additionally, a qualitative comparison of the algorithms can be made according to whether they are deterministic or probabilistic and the evaluation measures they employ. As it can be noticed, the acp is the most commonly used evaluation measure.

Deterministic methods

The LFC algorithm [10] achieves an acp value equal to 1, which means that all clusters are homogeneous. Simultaneously, the Rand index tends to zero and the misclassification rate is very low. In addition, the classification error reported is low for non-splitting speech segments [97–99]. Moreover, the step-by-step algorithm achieves low DER that is comparable to the corresponding values of the methods presented in [16,93,107]. The high performance of deterministic methods might be explained by the fact that they do not assume any model that describes the data.

Probabilistic methods

GMM-based methods
A straight comparison between GMM-based methods presented in [9,92] is possible since both of them employ the same evaluation measure (acp) and the same features on the same data set. It is obvious that [92] improves acp by 15%. The error in [103] is high, possibly due to underestimation of the number of clusters. Concerning the methods in [1,42], no direct comparison can be made. However, [42] achieves a high recall rate at a low false alarm rate.

HMM-based methods
The SID clustering algorithm achieves the highest acp compared to the other HMM-based algorithms. It also presents a high cluster coverage value and a low DER. The high classification error, reported in [95], could be due to the algorithm initialization and/or parameter settings. The lowest DER is achieved by [16,93,107] followed by [15]. The performance difference could be attributed to the different datasets and/or different features they employ.

3.6 Speaker clustering outlook

It has become clear that speaker segmentation results affect speaker clustering. The inclusion of speech that comes from two speakers in a single speech segment deteriorates speaker clustering performance. It is crucial speech segments to be homogeneous, which motivates the research community to strongly prefer over-segmentation. It complies with the fact that false alarms are considered as less cumbersome than miss detections.

In addition, non-speech and silence segments yield additional clusters, that might not be needed. Therefore, it is concluded that a preprocessing step that removes silence and non-speech segments is necessary.

Most of speaker clustering algorithms assume no prior knowledge on the number of clusters to be created. This fact usually leads to more clusters than the required ones. Of course, it is preferred to have more clusters, that can be merged in a latter step, than under-estimating the number of clusters.

It is worth mentioning that the complexity of a speaker clustering problem depends on the population size, the duration of the speech segment, the signal bandwidth, the environmental noise, the equipment, and whether the task has to be performed in real-time or not.

4 Conclusions

Due to the rapid increase in the volume of computer-available recorded speech, methods for speaker segmentation and clustering have been very appealing. Speaker segmentation has been a very active research topic in the past years. In this paper, several speaker segmentation algorithms have been reviewed and discussed. BIC-based speaker segmentation deficiencies are revealed along with methods to alleviate them. With respect to speaker clustering, various approaches have been presented and compared. It is established that speaker clustering efficiency depends on speaker segmentation results as well as the accurate estimation of the number of natural clusters. Speaker clustering techniques that deal effectively with the aforementioned inherent difficulties are presented in the paper. In conclusion, this survey has been compiled with the ambition to serve as either a starting point for researchers making their first steps in this exciting research area or to challenge mature researchers for further unification and standardization efforts revealing the vulnerable points of the existing literature. For example, the small sample case, the revision of Gaussianity assumption that is not always correct for modeling feature distributions, and the discrepancy between the actual and the estimated number of clusters are still open problems in speaker diarization. State-of-the-art speaker segmentation and clustering algorithms are sufficiently performing for tasks such as rich transcription or dialogue detection in clean conversations. However, there is still room for improvement, when conversations take place in noisy environments, such as during meeting recordings.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive criticism, that enabled them to improve the quality of their paper. This work has been supported by the "PYTHAGORAS II" Programme, funded in part by the European Union (75%) and in part by the Hellenic Ministry of Education and Religious Affairs (25%). M. Kotti was supported by the "Propondis" Public Welfare Foundation through scholarship.

References

[1] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, "Clustering speakers by their voices," in *Proc. 1998 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 757 - 760, Seattle, USA, May 1998.

[2] R. Sinha, S. E. Tranter, M. J. F. Gales, and P. C. Woodland, "The Cambridge

University March 2005 speaker diarisation system," in Proc. *European Conf. Speech Communication and Technology*, pp. 2437-2440, Lisbon, Portugal, September 2005.

[3] ISO/IEC 15938-4:2001, "Multimedia Content Description Interface - Part 4: Audio," Version 1.0.

[4] B. S. Manjunath, P. Salembier, T. Sikora, and P. Salembier, *Introduction to MPEG 7: Multimedia Content Description Language*. West Sussex, England: John Wiley & Sons, 2002.

[5] H. G. Kim and T. Sikora, "Comparison of MPEG-7 audio spectrum projection features and MFCC applied to speaker recognition, sound classification and audio segmentation," in Proc. *2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 5, pp. 925-928, Montreal, Canada, May 2004.

[6] H. G. Kim and T. Sikora, "Audio spectrum projection based on several basis decomposition algorithms applied togeneral sound recognition and audio segmentation," in Proc. *12th European Signal Processing Conf.*, pp. 1047-1050, Vienna, Austria, September 2004.

[7] M. Kotti, E. Benetos, and C. Kotropoulos, "Automatic speaker change detection with the Bayesian information criterion using MPEG-7 features and a fusion scheme," in Proc. *2006 IEEE Int. Symp. Circuits and Systems*, Kos, Greece, May 2006.

[8] M. Kotti, L. G. P. M. Martins, E. Benetos, J. S. Cardoso, and C. Kotropoulos, "Automatic speaker segmentation using multiple features and distance measures: A comparison of three approaches", in Proc. *2006 IEEE Int. Conf. Multimedia and Expo*, pp. 1101-1104, Toronto, Canada, July 2006.

[9] W. H. Tsai, S. S. Cheng, and H. M. Wang, "Speaker clustering of speech utterances using a voice characteristic reference space," in Proc. *Int. Conf. Spoken Language Processing*, Jeju Island, Korea, October 2004.

[10] D. Liu and F. Kubala, "Online speaker clustering," in Proc. *2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, pp. 333-336, Montreal, Canada, May 2004.

[11] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in Proc. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 645 - 648, Seattle, USA, May 1998.

[12] S. Meignier, J. F. Bonastre, and S. Igounet, "E-HMM approach for learning and adapting sound models for speaker indexing," in Proc. *Odyssey Speaker and Language Recognition Workshop*, pp. 175-180, Crete, Greece, June 2001.

[13] J. Ajmera, H. Bourlard, I. Lapidot, and I. McCowan, "Unknown-multiple speaker clustering using HMM," in Proc. *Int. Conf. Spoken Language Processing*, pp. 573-576, Colorado, USA, September 2002.

[14] X. Zhu, C. Barras, S. Meignier, and J.-L. Gauvain, "Combining speaker identification and BIC for speaker diarization," in Proc. *InterSpeech*, pp. 2441-2444, Lisbon, Portugal, September 2005.

[15] S. Meignier, D. Moraru, C. Fredouille, J. F. Bonastre, and L. Besacier, "Step-by-step and integrated approaches in broadcast news speaker diarization," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 303-330, April-July 2006.

[16] C. Barras, X. Zhu, S. Meignier, and J. L. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Trans. Audio, Speech, and Language Processing*, pp. 1505-1512, vol. 14, no. 5, September 2006.

[17] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech, and Language Processing*, pp. 1557-1565, vol. 14, no. 5, September 2006.

[18] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no 9, pp. 1437-1462, September 1997.

[19] V. Wan and W. M. Campbell "Support vector machines for speaker verification and identification," in Proc. *Neural Networks for Signal Processing*, vol. 10, pp. 775-784, Sydney, Australia, December, 2000.

[20] D. A. Reynolds, T. F. Quatiery, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19-41, October 2000.

[21] M. J. F. Gales, D. Y. Kim, P. C. Woodland, H. Y. Chan, D. Mrva, R. Sinha, and S. E. Tranter, "Progress in the CU-HTK broadcast news transcription system," *IEEE Trans. Speech and Audio Processing*, vol. 14, no 5, pp. 1513-1525, September 2006.

[22] National Institute of Standards and Technology (NIST) - The Segmentation Task: Find the Story Boundaries. `http://www.nist.gov/speech/tests/tdt/tdt99/presentations/NIST_segmentation/index.htm`

[23] The Center for Spoken Language Research of the Colorado University (CSLR). `http://cslr.colorado.edu/`

[24] International Computer Science Institute - Speech Research Group Berkeley. `http://www.icsi.berkeley.edu/groups/speech/`

[25] Speech Analysis and Interpretation Laboratory (SAIL) at the University of Southern California. `http://sail.usc.edu/projectsIntro.php`

[26] International Speech Technology and Research (STAR) Laboratory at Stanford research institute (SRI). `http://www.speech.sri.com/projects/sieve/`

[27] Microsoft Audio Projects. `http://research.microsoft.com/users/llu/Audioprojects.aspx`

[28] The Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP) Research Institute. `http://www.idiap.ch/speech_processing.php`

[29] The Laboratoire d'Informatique pour la Mècanique et les Sciences de l'Ingènieur (LIMSI) Spoken Language Processing Group. `http://www.limsi.fr/TLP`

[30] The Department of Speech, Music and Hearing of the Royal Institute of Technology (KTH) at Stockholm. `http://www.speech.kth.se`

[31] The Chair of Computer Science VI, Computer Science Department, Aachen University. http://www-i6.informatik.rwth-aachen.de

[32] The
Infant Speech Segmentation Project at Berkeley University. http://www-gse.berkeley.edu/research/completed/InfantSpeech.html

[33] Language Science Research Group, Washigton Univiversity. http://lsrg.cs.wustl.edu

[34] The University College of London Psychology Speech Group, speech segmentation issues. http://www.speech.psychol.ucl.ac.uk

[35] L. Lu and H. Zhang, "Speaker change detection and tracking in real-time news broadcast analysis," in Proc. *ACM Multimedia 2002*, pp. 602-610, Juan-les-Pins, France, December 2002.

[36] A. Tritschler and R. Gopinath, "Improved speaker segmentation and segments clustering using the Bayesian information criterion," in Proc. *6th European Conf. Speech Communication and Techology*, pp. 679-682, Budapest, Hungary, September 1999.

[37] P. Delacourt and C. J. Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing," *Speech Communication*, vol. 32, pp. 111-126, September 2000.

[38] S. Know and S. Narayanan, "Speaker change detection using a new weighted distance measure," in Proc. *Int. Conf. Spoken Language*, vol. 4, p. 2537-2540, Colorado, USA, September 2002.

[39] T. Wu, L. Lu, K. Chen, and H. Zhang, "UBM-based real-time speaker segmentation for broadcasting news," in Proc. *2003 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 193-196, Hong Kong, Hong Kong, April 2003.

[40] S. S. Cheng and H. M. Wang, "A sequential metric-based audio segmentation method via the Bayesian information criterion," in Proc. *8th European Conf. Speech Communication and Technology*, pp. 945-948, Geneva, Switzerland, September 2003.

[41] J. Ajmera, I. McCowan, and H. Bourlard, "Robust speaker change detection," *2004 IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 649-651, August 2004.

[42] L. Lu and H. Zhang, "Unsupervised speaker segmentation and tracking in real-time audio content analysis," *Multimedia Systems*, vol. 10, no. 4, pp. 332-343, April 2005.

[43] H. Kim, D. Elter, and T. Sikora, "Hybrid speaker-based segmentation system using model-level clustering," in Proc. *2005 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. I, pp. 745-748, Philadelphia, USA, March 2005.

[44] B. Zhou and J. H. L. Hansen, "Efficient audio stream segmentation via the combined T^2 statistic and the Bayesian information criterion," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 4, pp. 467-474, July 2005.

[45] S. Know and S. Narayanan, "Unsupervised speaker indexing using generic models," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 5, pp. 1004-1013, September 2005.

[46] C. H. Wu, Y. H. Chiu, C. J. Shia, and C. Y. Lin, "Automatic segmentation and identification of mixed-language speech using delta-BIC and LSA-based GMMs," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 266- 276, January 2006.

[47] L. Lu and H. Zhang, "Real-time unsupervised speaker change detection," in Proc. *16th Int. Conf. Pattern Recognition*, vol. 2, pp. 358-361, Quebec, Canada, August 2002.

[48] T. Wu, L. Lu, K. Chen, and H. Zhang, "Universal background models for real-time speaker change detection," in Proc. *9th Int. Conf. Multimedia Modeling*, pp. 135-149, Tamshui, Taiwan, January 2003.

[49] S. E. Tranter, K. Yu, G. Evermann, and P. C. Woodland, "Generating and evaluating segmentations for automatic speech recognition of conversational telephone speech," in Proc. *2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 433-477, Montreal, Canada, May 2004.

[50] D. Wang, L. Lu, and H. J. Zhang, "Speech segmentation without speech recognition", in Proc. *2003 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, pp. 468-471, Hong Kong, Hong Kong, April 2003.

[51] C. H. Wu and C. H. Hsieh, "Multiple change-point audio segmentation and classification using an MDL-based Gaussian model," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 647- 657, March 2006.

[52] H.-G. Kim, N. Moreau, and T. Sikora, *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. West Sussex, England: John Wiley & Sons, 2005.

[53] R. Huang and J. H. L. Hansen, "Advances in unsupervised audio segmentation for the broadcast news and ngsw corpora," in Proc. *2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 741-744, Montreal, Canada, May 2004.

[54] A.V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, N.J.: Prentice Hall, 1975.

[55] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Disrete-Time Processing of Speech Signals.* N.Y.: Wiley-IEEE, 1999.

[56] X. D. Huang, A. Acero, and H. -S. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper River Saddle, N.J.: Pearson Education - Prentice Hall, 2001.

[57] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", *J. Acoustical Society of America*, vol. 87, no. 4, pp. 1738-1752, April 1990.

[58] M. M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio and Electroacoustics*, vol. 16, no.2, pp. 262-266, June 1968.

[59] W. J. Hess, "Pitch and voicing determination," in *Advances in Speech Signal Processing*, S. Furui, M. M. Sondhi (Eds.), N.Y.: Marcel Dekker, Inc., 1991.

[60] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech Communication*, vol. 48, no. 9, pp. 1162-1181, September, 2006.

[61] B. Li, Y. Li, C. Wang, and C. Zhang, "A new efficient pitch-tracking algorithm," in Proc. *2003 IEEE Int. Conf. Robotics, Intelligent Systems and Signal Processing*, vol. 2, pp. 1102- 1107, Hunan, China, October 2003.

[62] N. Morgan and E. Fosler-Lussier, " Combining multiple estimators of speaking rate," in Proc. *1998 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 729 - 732, Seattle, USA, May 1998.

[63] M. Collet, D. Charlet, and F. Bimbot, "A correlation metric for speaker tracking using anchor models," in Proc. *2003 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, pp. 713-716, Hong Kong, Hong Kong, April 2003.

[64] B. L. Pellom and J. H. L. Hansen, "Automatic segmentation of speech recorded in unknown noisy channel characteristics", *Speech Communication*, vol. 25, no. 1-3, pp. 97-116, August 1998.

[65] J. Ajmera, I. McCowan, and H. Bourland, "Speech/music segmentation using entropy and dynamism features in a HMM classification framework", *Speech Communication*, vol. 40, no.3, pp. 351-363, May 2003.

[66] N. Mesgarani, S. Shamma, and M. Slaney, "Speech discrimination based on multiscale spectro-temporal modulations", in Proc. *2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, pp. 601-604, Montreal, Canada, May 2004.

[67] J. A. Arias, J. Pinquier, and R. Andè-Obrecht, "Evaluation of classification techniques for audio indexing," in Proc. *13th European Signal Processing. Conf.*, Antalya, Turkey, September 2005.

[68] H. Harb and L. Chen, "Audio-based description and structuring of videos," *Int. J. Digital Libraries*, vol. 6, no. 1, pp. 70-81, February 2006.

[69] T . Kemp, M. Schmidt, M. Westphal, and A., Waibel, "Strategies for automatic segmentation of audio data," in Proc. *2000 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp. 14231426, Istanbul, Turkey, June 2000.

[70] F. Bimbot, I. Magrin-Chagnolleau, and L. Mathan, "Second-order statistical measures for text-independent speaker identification," *Speech Communication*, vol. 17, no. 1-2, pp. 177-192, August 1995.

[71] J. H. L. Hansen, R. Huang, B. Zhou, M. Seadle, J. R. Deller, A. R. Gurijala, M. Kurimo, and P. Angkititrakul, "SpeechFind: Advances in spoken document retrieval for a national gallery of the spoken word," *IEEE Trans. Speech and Audio Processing*, vol. 13, no 5, pp. 712-730, September 2005.

[72] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion", in Proc. *DARPA Broadcast News Transcription Understanding Workshop*, pp. 127-132, Landsdowne, VA, February 1998.

[73] M. Cettolo and M. Vescovi, "Efficient audio segmentation algorithms based on the BIC", in Proc. *2003 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, pp. 537-540, Hong Kong, Hong Kong, April 2003.

[74] M. Vescovi, M. Cettolo, and R. Rizzi, "A DP algorithm for speaker change detection", in Proc. *8th European Conf. Speech Communication and Technology*, pp. 2997-3000, Geneva, Switzerland, September 2003.

[75] Q. Jin, K. Laskowski, T. Schultz, and A.Waibel, "Speaker segmentation and clustering in meetings," in Proc. *NIST Meeting Recognition Workshop*, pp. 112-117, Montreal, Canada, May 2004.

[76] M. Cettolo, M. Vescovi, and R. Rizzi, "Evaluation of BIC-based algorithms for audio segmentation," *Computer Speech and Language*, vol. 19, pp. 1004-1013, April 2005.

[77] N. A. Campbell, "Robust procedures in multivariate analysis I: Robust covariance estimation", *Applied Statistics*, vol. 29, no. 3, pp. 231-237, 1980.

[78] S. Cheng and H. Wang, "Metric SEQDAC: A hybrid approach for audio segmentation," in Proc. *8th Int. Conf. Spoken Language Processing*, pp. 1617-1620, Jeju, Korea, October 2004.

[79] H. Gish, M. H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification", in Proc. *1991 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 873-876, Toronto, Canada, April 1991.

[80] J. Ajmera, G. Lathoud, I. McCowan, "Clustering and segmenting speakers and their locations in meetings," in Proc. *2004 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol 1, pp. 605-608, Montreal, Canada, May 2004.

[81] D. P. W. Ellis and J.C. Liu, "Speaker turn segmentation based on between-channel differences," in Proc. *NIST Meeting Recognition Workshop*, pp. 112-117, Montreal, Canada, May 2004.

[82] J. Alabiso, R. MacIntyre, and D. Graff, "1997 English Broadcast News Transcripts (HUB4)," Linguistic Data Consortium, Philadelphia, 1998.

[83] D. Graff, "TDT3 Mandarin Audio," Linguistic Data Consortium, Philadelphia, 2001.

[84] M. Federica, D. Giordani, and P. Caletti, "Development and evaluation of an Italian broadcast news corpus", in Proc. *2nd Int. Conf. Language Resources and Evaluation*, pp. 921924, Athens, Greece, May-June 2000.

[85] H. M. Wang, B. Chen, J. W. Kuo, and S. S. Cheng, "MATBN: A mandarin chinese broadcast news corpus," *Computational Linguistics and Chinese Language Processing*, vol. 10, no. 2, pp. 219-236, June 2005.

[86] Y. Zhu and X. Rong, "Unified fusion rules for multisensor multihypothesis network decision systems," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 33, no.4, pp. 502-513, July 2003.

[87] J. S. Garofolo, "DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus," Linguistic Data Consortium, Philadelphia, 1993.

[88] J. J. Godfrey and E. Holliman, "Switchboard-1 Release 2", Linguistic Data Consortium, Philadelphia, 1997.

[89] M. Kotti, E. Benetos, and C. Kotropoulos, "Computationally efficient and robust BIC-based speaker segmentation," *IEEE Trans. Audio, Speech, and Language Processing*, in revision.

[90] The Linguistic Data Consortium. `http://www.ldc.upenn.edu/`

[91] G. Almpanidis and C. Kotropoulos, "Phonemic segmentation using the generalised Gamma distribution and small sample Bayesian information criterion," *Speech Communication*, vol. 50, no.1, pp. 38-55, January 2008.

[92] W.-H. Tsai and H.-M. Wang, "Speaker clustering of unknown utterances based on maximum purity estimation," in *Proc. European Conf. Speech Communication and Technology*, Lisbon, Portugal, September 2005.

[93] J.-L. Gauvain, L. Lamel, and G. Adda, "Partitioning and transcription of broadcast news data", in *Proc. Int. Conf. Spoken Language Processing,* pp. 1335-1338, Sydney, Australia, December 1998.

[94] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, N.J.:Prentice Hall, 1988.

[95] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in Proc. *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 411-416, Virgin Islands, November 2003.

[96] I. Voitovetsky, H. Guterman, and A. Cohen, "Validity criterion for unsupervised speaker recognition", in *Proc. First Workshop Text, Speech, and Dialogue*, pp. 321-326, Brno, Czech Republic, September 1998.

[97] I. Voitovetsky, H. Guterman, and A. Cohen, "Unsupervised speaker classification using self-organizing maps," in *Proc. IEEE Workshop Neural Networks for Signal Processing*, pp. 578-587, Amelia Island, USA, September 1997.

[98] I. Lapidot and H. Guterman, "Resolution limitation in speakers clustering and segmentation problems," in Proc. *2001: A Speaker Odyssey, The Speaker Recognition Workshop*, pp. 169-173, Chania, Greece, June 18-22, 2001.

[99] I. Lapidot, H. Guterman, and A. Cohen, "Unsupervised speaker recognition based on competition between self-organizing maps," *IEEE Trans. Neural Networks*, vol. 13, no. 4, pp. 877-887, July 2002.

[100] H. Bozdogan, "Akaike's Information Criterion and Recent Developments in Information Complexity," *Mathematical Psychology Archive*, vol. 44 , no. 1, pp. 62-91, March 2000.

[101] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. 2/e. N.Y.:John Wiley & Sons, 2001.

[102] M. Przybocki and A. Martin, "2001 NIST Speaker Recognition Evaluation Corpus", Linguistic Data Consortium, Philadelphia,

[103] H. Jin, F. Kubala, and R. Schwartz, "Automatic speaker clustering," in *Proc. Speech Recognition Workshop*, pp. 108-111, Chantilly, Virginia, 1997.

[104] D. Graff and J. Alabiso, "1996 English Broadcast News Transcripts (HUB4)", Linguistic Data Consortium, Philadelphia, 1997.

[105] 1998 HUB4 Broadcast News Evaluation English Test Material, Linguistic Data Consortium, Philadelphia, 2000.

[106] J. Fiscus, J. Garofolo, M. Przybocki, W. Fisher, and D. Pallett, "1997 English Broadcast News Speech (HUB4)", Linguistic Data Consortium, Philadelphia, 1998.

[107] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Improving speaker diarization," in *Proc. Fall Rich Transcription Workshop (RT-04)*, Palisades, N.Y, November 2004, [Online]. Available: `http://www.limsi.fr/Individu/barras/publis/rt04f_diarization.pdf`.

[108] D. Graff, J. Alabiso, J. Fiscus, J. Garofolo, W. Fisher, and D. Pallett, "1996 English Broadcast News Dev and Eval (HUB4)", Linguistic Data Consortium, Philadelphia, 1997.

[109] V. Moschou, M. Kotti, and C. Kotropoulos, "A novel speaker diarization system with application to movie scene analysis", in *2008 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, submitted.