Engineering Applications of Artificial Intelligence 22 (2009) 667–675

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Speaker diarization using autoassociative neural networks

S. Jothilakshmi *, V. Ramalingam, S. Palanivel

Department of Computer Science and Engineering, Annamalai University, Annamalainagar 608 002, India

ARTICLE INFO

Article history:
Received 3 October 2008
Received in revised form
2 January 2009
Accepted 29 January 2009
Available online 9 March 2009

Keywords:
Speaker diarization
Speaker segmentation
Speaker clustering
Mel frequency cepstral coefficients
Autoassociative neural networks

ABSTRACT

This paper addresses a new approach to speaker diarization using autoassociative neural networks (AANN). The speaker diarization task consists of segmenting a conversation into homogeneous segments which are then clustered into speaker classes. The proposed method uses AANN models to capture the speaker specific information from mel frequency cepstral coefficients (MFCC). The distribution capturing ability of the AANN model is utilized for segmenting the conversation and grouping each segment into one of the speaker classes. The algorithm has been tested on different databases, and the results are compared with the existing algorithms. The experimental results show that the proposed approach competes with the standard speaker diarization methods reported in the literature and it is an alternative method to the existing speaker diarization methods.

1. Introduction

Speaker diarization is the process of automatically partitioning a conversation involving multiple speakers into homogeneous segments and grouping together all the segments that correspond to the same speaker. The first part of the process is known as speaker segmentation or speaker change detection while the second one is called as speaker clustering. Hence speaker change detection followed by speaker clustering is known as speaker diarization (Meignier et al., 2006; Tranter and Reynolds, 2006; Sinha et al., 2005).

Nowadays a rapid increase in the volume of recorded speech is manifested which includes television and audio broadcasts, voice mails, meeting and other spoken documents (Solomonoff et al., 1998; Kotti et al., 2007). There is a growing need to apply automatic human language technologies to allow efficient and effective searching, indexing and accessing of these information sources. Diarization can be used for helping speech recognition, facilitating the searching and indexing of audio archives and increasing the richness of automatic transcriptions, making them more reliable and potentially helping with other tasks such as summarization, parsing and machine translation (Tranter and Reynolds, 2006).

Generally, for the task of speaker diarization no prior information is available regarding the number of speakers involved or their identities. So, speaker diarization can be considered as a task of identifying the number of speakers and creating a list of speech time intervals for each speakers. In the literature, various speaker diarization algorithms have been proposed. These algorithms can be categorized into three categories: step by step approaches, integrated approaches and mixed approaches.

Step by step approaches divide the speaker diarization task into number of steps (Siu et al., 1992; Wilcox et al., 1994; Sieglar et al., 1997; Gauvain et al., 1998; Chen and Gopalakrishnan, 1998). First finding the speaker change points using the symmetric Kullback Leibler (KL2), the generalized likelihood ratio (GLR) or the Bayesian information criterion (BIC) distance approaches, then growing the segments during a hierarchical clustering phase and finally determining the number of speakers. In the case of integrated approaches (Meigneir et al., 2001; Ajmera and Wooters, 2003) all the steps involved in speaker diarization are performed simultaneously. Mixed strategies also proposed in Wilcox et al. (1994), Moraru et al. (2004), and Reynolds et al. (2000), where classical step by step segmentation and clustering are first applied and then refined using a re-segmentation process during which the segment boundaries, the segment clustering and sometimes the number of speakers are refined.

Most of the model based speaker diarization systems in the literature use Gaussian mixture model (GMM) or hidden Markov model (HMM) to estimate the probability distribution of the feature vectors of a speaker. While GMMs appear to be general enough to characterize the distribution of the given data, the model is constrained by the fact that the shape of the components of the distribution is assumed to be Gaussian, and the number of mixtures are fixed a priori (Yegnanarayana and Kishore, 2002).

* Corresponding author. Tel.: +91 9894 693493; fax: +91 4144 238080.
E-mail addresses: jothi.sekar@gmail.com, jothi_sekar1993@yahoo.com (S. Jothilakshmi), aucsevr@yahoo.com (V. Ramalingam), spal_yughu@yahoo.com (S. Palanivel).

0952-1976/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2009.01.012

In this context, Yegnanarayana and Kishore (2002) investigated the potential of nonlinear models such as autoassociative neural network (AANN) models, which perform identity mapping of the input space. AANN is a feed forward neural network which can be designed to perform the task of pattern classification or pattern mapping (Yegnanarayana, 1999).

The main contribution of this paper concerns the use of the distribution capturing ability of the AANN for speaker change detection and speaker clustering for speaker diarization. The proposed method relies on a classical two step speaker diarization approach based on a detection of speaker turns followed by a clustering process as shown in Fig. 1. This work formulates a new speaker diarization algorithm and it works without any prior knowledge of the identity of speakers.

The rest of the paper is organized as follows: A brief description about the method of extracting speaker specific information from the speech signal is described in Section 2. AANN model for capturing the distribution of acoustic feature vectors is given in Section 3. The proposed algorithm for speaker diarization is presented in Section 4. In Section 5, the performance measures used for speaker diarization are discussed. Section 6 presents the experimental results and the performance comparison of the proposed method with the existing methods. Section 7 concludes the paper.

2. Feature extraction for speaker segmentation

Mel frequency cepstral coefficients (MFCC) have proved to be one of the most successful feature representations in speech related recognition tasks. The mel-cepstrum exploits auditory principles, as well as the decorrelating property of the cepstrum (Davis and Mermelstein, 1980). In this section we briefly describe the signal processing involved in extracting the MFCC. The selected properties for the speech signals are a sampling rate of 8 kHz, 16 bit monophonic, pulse code modulation (PCM) format in wav audio. The procedure of MFCC computation is shown in Fig. 2 and described as follows:

- *Preemphasis*: The digitized speech signal $s(n)$ is put through a low order digital system to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the signal processing. The output of the preemphasis network, $\hat{s}(n)$ is related to the input $s(n)$ by the difference equation

$$\hat{s}(n) = s(n) - \alpha s(n-1) \tag{1}$$

The most common value for α is around 0.95.

- *Frame blocking*: Speech analysis usually assumes that the signal properties change relatively slowly with time. This allows examination of a short time window of speech to extract parameters presumed to remain fixed for the duration of the window. Thus to model dynamic parameters, we must divide the signal into successive windows or analysis frames, so that the parameters can be calculated often enough to follow the relevant changes. In this step the preemphasized speech signal, $\hat{s}(n)$ is blocked into frames of N samples, with adjacent frames being separated by M samples. If we denote the lth frame speech by $x_l(n)$, and there are L frames within the entire speech signal, then

$$x_l(n) = \hat{s}(Ml+n), \quad n = 0,1,\ldots,N-1, \;\; l = 0,1,\ldots,L-1 \tag{2}$$

We used a frame rate of 125 frames/s, where each frame was 16 ms in duration with an overlap of 50% between adjacent frames.

- *Windowing*: The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of the frame. The window must be selected to taper the signal to zero at the beginning and end of each frame. If we define the window as $w(n)$, $0 \leqslant n \leqslant N-1$, then the result of windowing the signal is

$$\tilde{x}_l(n) = x_l(n)w(n), \quad 0 \leqslant n \leqslant N-1 \tag{3}$$

The Hamming window is used for our work, which has the form

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leqslant n \leqslant N-1 \tag{4}$$

- *Computing spectral coefficients*: The spectral coefficients of the windowed frames are computed using Fast Fourier Transform, as follows:

$$X(k) = \sum_{n=0}^{N-1} \tilde{x}_l(n)\exp^{-jk(2\pi/N)n}, \quad 0 \leqslant n \leqslant N-1 \tag{5}$$

- *Computing mel spectral coefficients*: The spectral coefficients of each frame are then weighted by a series of filter frequency responses whose center frequencies and bandwidths roughly match those of the auditory critical band filters. These filters

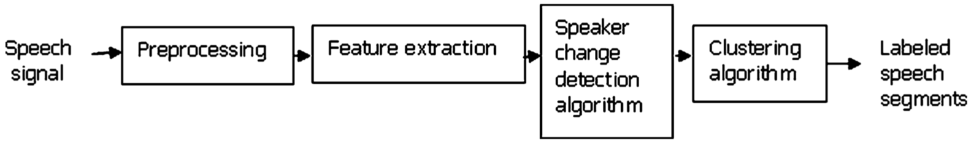

Fig. 1. Block diagram of speaker diarization system.

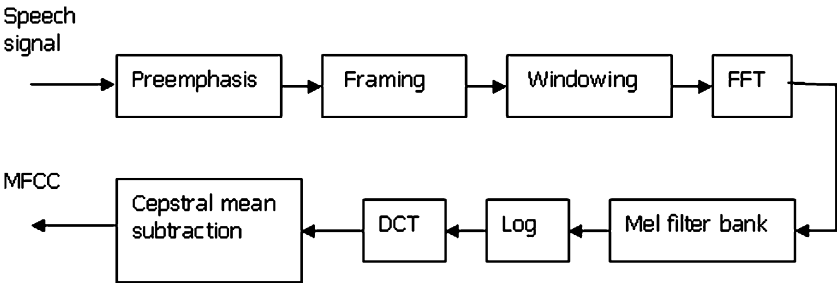

Fig. 2. Extraction of MFCC from speech signal.

follow the mel scale whereby band edges and center frequencies of the filters are linear for low frequency and logarithmically increase with increasing frequency. We call these filters as mel-scale filters and collectively a mel-scale filter bank. As can be seen, the filters used are triangular and they are equally spaced along the mel scale which is defined by

$$\mathrm{Mel}(f) = 2595\log_{10}\left(1+\frac{f}{700}\right) \tag{6}$$

Each short term Fourier transform (STFT) magnitude coefficient is multiplied by the corresponding filter gain and the results are accumulated.

- *Computing MFCC*: The discrete cosine transform (DCT) is applied to the log of the mel spectral coefficients to obtain the MFCC as follows:

$$x(m) = \sqrt{\frac{2}{M}}\sum_{i=0}^{M-1} E(i)\cos\left(\frac{(2x+1)m\pi}{2N}\right), \quad m = 1,\ldots,M \tag{7}$$

where M is the number of filters in the filter bank. We used the first 19 MFCC, other than the zeroth value to evaluate our algorithm. Cepstral mean subtraction is performed to reduce the channel effects.

3. AANN model for capturing the distribution of acoustic feature vectors

AANN models are feed forward neural networks performing an identity mapping of the input space, and are used to capture the distribution of the input data (Yegnanarayana and Kishore, 2002; Palanivel, 2004). Limitation of PCA to represent an input space using a linear subspace motivated the researchers to investigate a method of projecting the input data onto a nonlinear subspace using AANN models (Kramer, 1991; Bourlard and Kamp, 1988). A three layer AANN consists of three layers namely input layer, hidden layer and output layer. An AANN is a feed forward network with the desired output being the same as the input vector. Therefore, the number of units in the input and output layers is equal. The number of hidden layers and the number of units in each hidden layer depend on the problem. A three layer AANN model clusters the input data in the linear subspace, whereas a five layer AANN model captures the nonlinear subspace passing through the distribution of the input data. Studies on three layer AANN models show that the nonlinear activation function at the hidden units clusters the input data in a linear subspace (Bianchini et al., 1995). Theoretically, it was shown that the weights of the network will produce small errors only for a set of points around the training data (Bianchini et al., 1995). When the constraints of the network are relaxed in terms of layers, the network is able to cluster the input data in the nonlinear subspace. Hence a five layer AANN model as shown in Fig. 3 is used to capture the distribution of the feature vectors in our study.

Let us consider the five layer AANN model shown in Fig. 3, which has three hidden layers. The processing units in the first and third hidden layers are nonlinear, and the units in the second compression/hidden layer can be linear or nonlinear. As the error between the actual and the desired output vectors is minimized, the cluster of points in the input space determines the shape of the hyper surface obtained by the projection onto the lower dimensional space. The second and fourth layers of the network have more units than the input layer. The third layer has fewer units than the first or fifth. The activation functions at the second, third and fourth layers are nonlinear. The structure of the AANN model used in our study is 19L 38N 5N 38N 19L, where L denotes a linear unit and N denotes a nonlinear units. The nonlinear output function for each unit is tanh(s), where s is the activation value of the unit. The standard back propagation learning algorithm (Haykin, 1999) is used to adjust the weights of the network to minimize the mean square error for each feature vector. The AANN captures the distribution of the input data depending on the constraints imposed by the structure of the network, just as the number of mixtures and Gaussian functions do in the case of GMM. The choice of parameters such as feature vectors, initial weights and structure of AANN is not very critical, as variation of these parameters does not affect the performance of the system abruptly (Kishore, 2000).

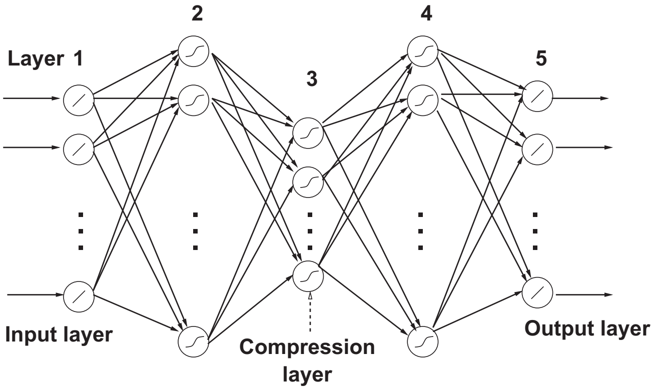

Fig. 3. A five layer AANN model.

In order to visualize the distribution better, one can plot the error for each input data point in the form of some probability surface as shown in Fig. 4. The error e_i for the data point i in the input space is plotted as $p_i = \exp^{(-e_i/\alpha)}$, where α is a constant. Note that p_i is not strictly a probability density function, but we call the resulting surface as probability surface. The plot of the probability surface shows a large amplitude for smaller error e_i, indicating better match of the network for that data point. The constraints imposed by the network can be seen by the shape of the error surface takes in both the cases. One can use the probability surface to study the characteristics of the distribution of the input data captured by the network. Ideally, one would like to achieve the best probability surface, best defined in terms of some measure corresponding to a low average error.

During AANN training, the weights of the network are adjusted to minimize the mean square error obtained for each feature vector. If the adjustment of weights is done for all feature vectors once, then the network is said to be trained for one epoch. For successive epochs, the mean square error averaged over all feature vectors. During testing phase, the features extracted from the test data are given to the trained AANN model to obtain the average error.

4. The proposed speaker diarization algorithm

The speaker diarization involves the technical elements presented in the previous section, and two main steps: speaker change detection and speaker clustering. It is assumed that acoustic features have been extracted from the speech signal.

The outline of the algorithm is shown in Figs. 5 and 6 and summarized as follows: After obtaining the speech features for each frame of the given conversation, initially a block of frames are selected starting from the first frame and it is assumed that the speaker change occurs at the middle frame of the block. AANN model is created to capture the distribution of left half of the block [LHB]. The feature vectors of the right half of the block [RHB] are used for testing the model. If speaker change occurs at the middle frame (i.e.,) RHB and LHB will be from different speakers and all

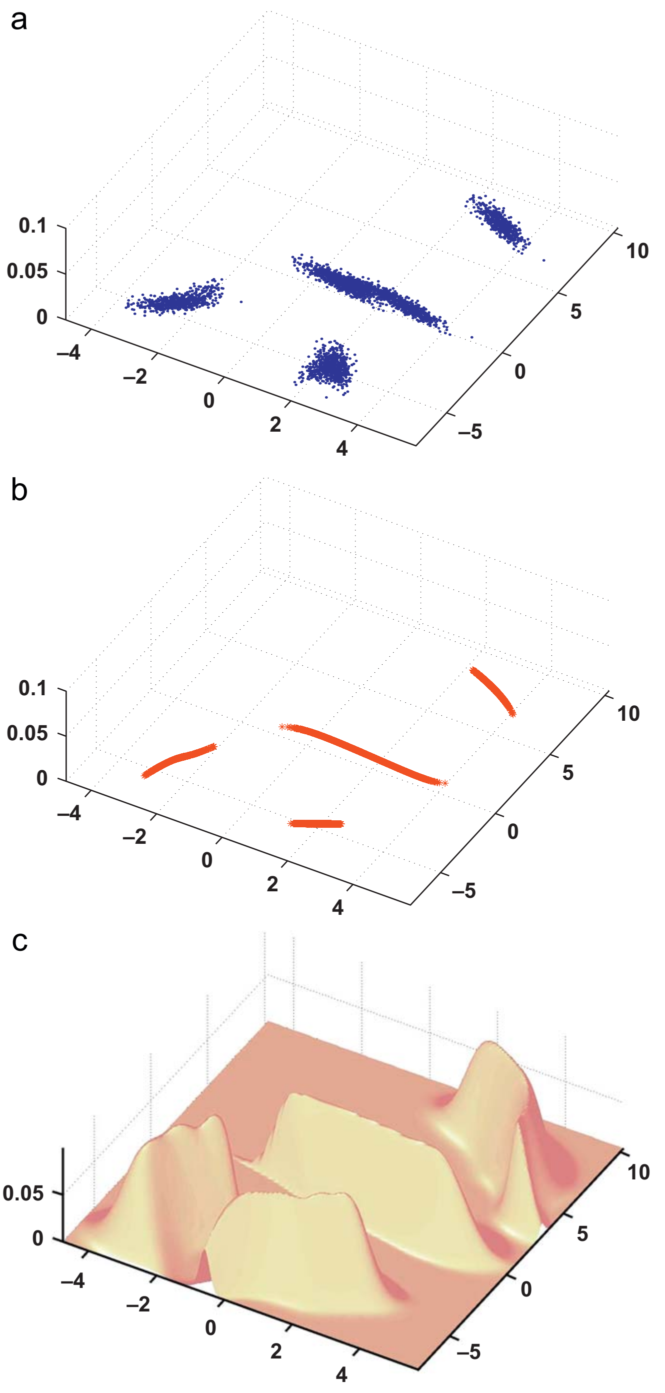

Fig. 4. Distribution capturing ability of AANN model. (a) Artificial two-dimensional data. (b) Two-dimensional output of AANN model. (c) Probability surfaces realized by the network.

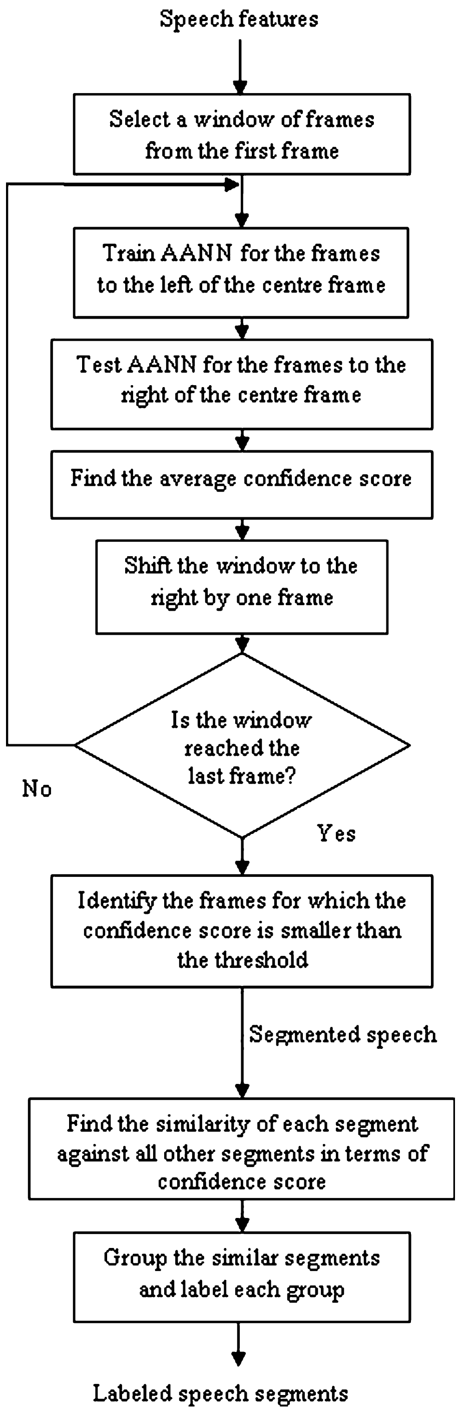
Speech features
Select a window of frames from the first frame
Train AANN for the frames to the left of the centre frame
Test AANN for the frames to the right of the centre frame
Find the average confidence score
Shift the window to the right by one frame
Is the window reached the last frame?
No
Yes
Identify the frames for which the confidence score is smaller than the threshold
Segmented speech
Find the similarity of each segment against all other segments in terms of confidence score
Group the similar segments and label each group
Labeled speech segments

Fig. 5. Flow chart of the proposed speaker diarization system.

the feature vectors from the RHB may not fall into the distribution and the model gives low confidence (probability) score. Likewise, if the middle frame is not the true speaker change point and both LHB and RHB are from the same speaker then the confidence score of RHB will be very high. The next possibility is either LHB or RHB may have the speech features from both the speakers. If this is the case, the confidence score of RHB will be in between the above two values. After obtaining the confidence score for this middle frame, the block is shifted by one frame to the right. Then the entire procedure is repeated for this new block and the confidence score is obtained by assuming the middle frame of this new block as speaker change point. Likewise the confidence scores are obtained until RHB reached the last frame of the speech frames. From the confidence score, the local minima positions are the speaker change points and they are detected using a threshold. After detecting the speaker changes, the segments obtained must be clustered to determine the number of speakers.

4.1. Speaker change detection

We begin with the assumption that there is a speaker change located in the data stream at the center of the analysis window under consideration. If the speech signal of this analysis window comes from different speakers, all the feature vectors in the right

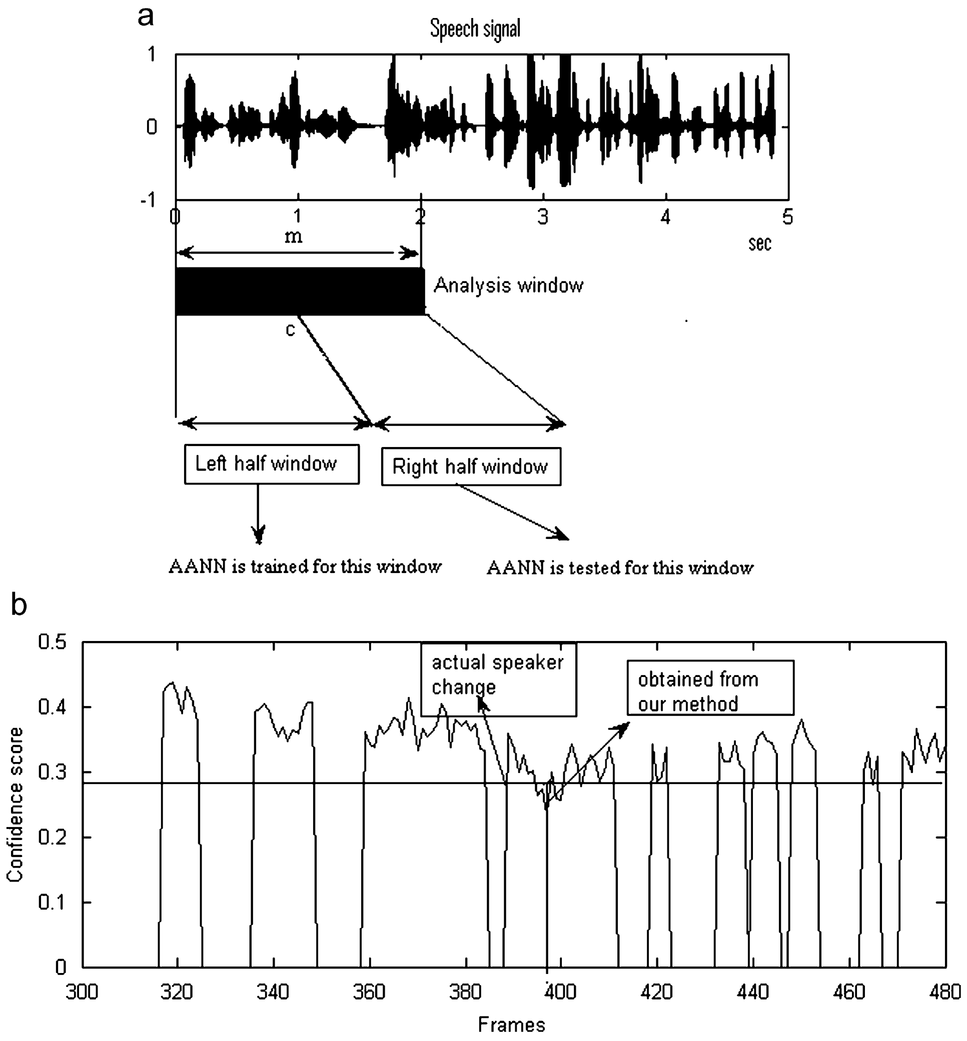

Fig. 6. Concept of the proposed segmentation algorithm. (a) Speech signal. (b) Confidence score.

half of the window may not fall into the distribution of the feature vectors from the left half window. On the contrary, if the speech signal of this analysis window comes from only one speaker the feature vectors in the right half of the window falls into the distribution of feature vectors of the left half window.

Given the speech features $S = \{s_i : i = 1, 2, \ldots, n\}$ where i is the frame index and n is the total number of frames in the speech signal. The proposed algorithm for detecting speaker change is summarized as follows:

(1) From n frames, m number of frames are selected such that $m \bmod 2 = 1$, and considered as analysis window W_k. W_k is the kth analysis window which is given by

$$W_k = \{s_j\}, \quad k \leqslant j < m + k \tag{8}$$

(2) It is assumed that the speaker change occurs at the middle frame (c) of the analysis window

$$c = k + \left\lfloor \frac{m}{2} \right\rfloor \tag{9}$$

(3) We consider all the frames in the analysis window that are located left of c as left half window L_k

$$L_k = \{s_j\}, \quad k \leqslant j < c - 1 \tag{10}$$

At the same time, we consider all the frames that are located right of c as right half window R_k

$$R_k = \{s_j\}, \quad c + 1 \leqslant j < m + k \tag{11}$$

(4) AANN is trained using the frames in L_k and the model captures the distribution of this block of data. Then the feature vectors in R_k are given as input to the AANN model and the output of the model is compared with the input to compute the normalized squared error e_k. The normalized squared error (e_k) for the feature vector $\boldsymbol{y}$ is given by

$$e_k = \frac{\|\boldsymbol{y} - \boldsymbol{o}\|^2}{\|\boldsymbol{y}\|^2} \tag{12}$$

where $\boldsymbol{o}$ is the output vector given by the model. The error e_k is transformed into a confidence score s using

$$s = \exp(-e_k) \tag{13}$$

The average confidence score is calculated by summing the confidence score of the individual frames and the result is divided by the number of frames in the block. We tried with the weighted summation of the frame scores within the block and there is no improvement in the performance. If true speaker change occurs at c, then L_k and R_k will be from different speakers and the average confidence score for this c

will be very low. Likewise, if c is not the true speaker change point and both L_k and R_k are from the same speaker then the average confidence score will be very high. The next possibility is that either L_k or R_k may have the speech features from both the speakers. If this is the case, the average confidence score will be in between the above two values.

(5) The value of k is incremented by one and the steps from 1 to 4 are repeated until $m+k$ reaches n.

(6) Detect the speaker change points from the confidence score by applying a threshold. The threshold (t_s) is calculated from the confidence score as follows:

$$t_s = s_{min} + a s_{min}, \quad 0 < a < 1 \tag{14}$$

where s_{min} is the global minimum confidence score and a is the adjustable parameter.

The thresholding step is performed as in any other detection algorithm: the threshold is tuned in accordance to some trade off between false alarms (FA) and missed detections. The research community tends to treat FA as less cumbersome than missed detections. Because over segmentation caused by a high number of FA is easier to remedy than under-segmentation caused by high number of miss detections. In our algorithm when a is nearer to 0 the number of miss detections will be more, if it is nearer to 1 the number of FA will be more. So the parameter a is selected to achieve over segmentation. As the clustering step follows the segmentation step, the false alarmed segments will be clustered to same speaker group due to similar speech characteristics.

4.2. Speaker clustering

Once the speaker change points have been detected, the next important step is clustering. It consists of labeling segments of speech, detected by the speaker change detection algorithm given in previous subsection, with speaker labels. Let the speaker segments $C = \{c_i : i = 1, 2, \ldots, P\}$ where i is the segment index and P is the total number of segments obtained from the segmentation algorithm. The proposed algorithm for speaker clustering is described as follows:

(1) For each segment c_i, the AANN is trained using the frames in c_i and the model captures the distribution of this block of data. Then the feature vectors of each segment c_j is given as input to the AANN model and the output of the model is compared with the input to compute the confidence score s_{ij} which is the confidence score of jth test segment against ith trained segment. From the outcome of this step, a confidence score matrix s_{mat} of size $P \times P$ can be formed such that

$$s_{mat} = \begin{pmatrix} s_{11} & s_{12} & \ldots & s_{1P} \\ s_{21} & s_{22} & \ldots & s_{2P} \\ & \vdots & & \\ s_{P1} & s_{P2} & \ldots & s_{PP} \end{pmatrix}$$

(2) As both s_{ij} and s_{ji} denotes the similarity or confidence score between the segments i and j, hence s_{ik} can be calculated using

$$s_{ik} = \begin{cases} (s_{ij} + s_{ji})/2 & \text{if } k \geqslant i \\ 0 & \text{if } k < i \end{cases} \tag{15}$$

Now s_{mat} becomes

$$s_{mat} = \begin{pmatrix} s_{11} & s_{12} & \ldots & s_{1P} \\ 0 & s_{22} & \ldots & s_{2P} \\ & \vdots & & \\ 0 & 0 & \ldots & s_{PP} \end{pmatrix}$$

(3) The distances between ith segment and all other segments are computed by

$$d_{ik} = s_{ii} - s_{ik}, \quad k > i \tag{16}$$

Now s_{mat} becomes

$$s_{mat} = \begin{pmatrix} s_{11} & d_{12} & \ldots & d_{1P} \\ 0 & s_{22} & \ldots & d_{2P} \\ & \vdots & & \\ 0 & 0 & \ldots & s_{PP} \end{pmatrix}$$

(4) If

$$d_{ik} < s_{ii} t_c, \quad 0 < t_c < 0.5, \quad 0 < i \leqslant P, \quad i < k \leqslant P \tag{17}$$

ith segment and kth segment are more similar in characteristics. So, both these segments are belonging to same speaker class and can be clustered together where t_c is the cluster parameter. If ith segment is already grouped to any one of the speaker class, kth segment also grouped to that speaker class or a new speaker class is formed with ith segment, and the kth segment will be grouped to that speaker class. Number of speaker classes will be equal to the number of speakers in the conversation.

5. Performance measures

The performance of speaker segmentation is assessed in terms of two types of error related to speaker change detections namely FA and missed detections. A FA (α) of speaker change detection occurs when a detected speaker change is not a true one. A missed detection (β) occurs when a true speaker change cannot be detected. The FA rate (α_r) and missed detection rate (β_r) are defined as (Delacourt and Wellekens, 2000; Cheng and Wang, 2004)

$$\alpha_r = \frac{\textit{Number of false alarms}}{\textit{Number of actual speaker changes} + \textit{Number of false alarms}} \tag{18}$$

$$\beta_r = \frac{\textit{Number of missed detections}}{\textit{Number of actual speaker changes}} \tag{19}$$

To compute these different metrics, it is necessary to take into account that the position of the speaker turns are not exactly defined, due to the presence of inter speaker silences or nonspeech sounds. Therefore, it is considered that a changing point is correctly located if it belongs to a time interval $[t_0 - \Delta t, t_0 + \Delta t]$ in which t_0 is the reference mark and Δt is the tolerance. In our case the tolerance is 0.25 s.

The clustering step is evaluated in terms of average cluster purity (acp) and average speaker purity (asp) as defined in Kotti et al. (2007). Where acp is a measure of how well a cluster is limited to only one speaker and the asp describes how well a speaker is limited to only one cluster.

In order to evaluate the performance of overall diarization, we used the evaluation metric called *diarization error rate* (*DER*) (NIST, 2004; Tranter and Reynolds, 2006; Kotti et al., 2007). 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 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

$$DER = MS + FA + SE \tag{20}$$

6. Experiments and results

This section presents experimental results using different speech databases.

6.1. The databases

There are three primary domains which have been used for speaker diarization research and development: broadcast news, recorded meetings and telephone conversations. We experimented our algorithm only on broadcast news. Two different types of speech data have been used to test the performance of the algorithm. Initially the experiments have been performed over a corpus which is composed of 16 broadcast news of about 30 min each recorded from various channels like BBC, NDTV and SUN News. In this paper this corpus is named as *AUdata*. Then the experiments have been carried out using NIST-RT'03S speaker diarization evaluation on American broadcast news. The dataset is divided into development corpus and evaluation corpus. The development corpus is used for training the system and tuning the parameters which is composed of six broadcast news shows of about 10 min each. The evaluation corpus is composed of three 30 minutes show recorded from various American channels which is for validation.

6.2. Feature extraction

We used the first 19 MFCC, other than the zeroth value to evaluate our algorithm. Cepstral mean subtraction is performed to reduce the channel effects. The selected properties for the speech signals are a sampling rate of 8 kHz, 16 bit monophonic PCM format. We used a frame rate of 125 frames/s, where each frame is 16 ms in duration with an overlap of 50% between adjacent frames.

6.3. Parameter tuning phase

In order to tune the parameters of the algorithm that yield best performance, several experiments have been conducted on AUdata, whose results are reported in this subsection. The parameters to be tuned are: the size of the analysis window, number of epochs (one epoch of training is a single presentation of all the training vectors to the network), adjustable parameter a and cluster parameter t_c. The MFCC feature vectors are extracted for all the speech frames as described in Sections 2 and 6.2. For each analysis window, the distribution of the feature vectors is captured using the AANN model as described in Section 3. The feature vectors of R_k are given as input to the AANN model and the average confidence score is calculated as described in Section 4.1. Fig. 7 shows the confidence score obtained for analysis window size of 65, 95, 125 and 140 frames. In this experiment, the number of epochs is 100 and the adjustable parameter a is 0.5. The number of FA and miss detections are significantly low for the 125 frames block size when compared to analysis window size settings of 65, 95 and 140. So, in this work we used the analysis window size of 125 frames. Moreover the window size of 125

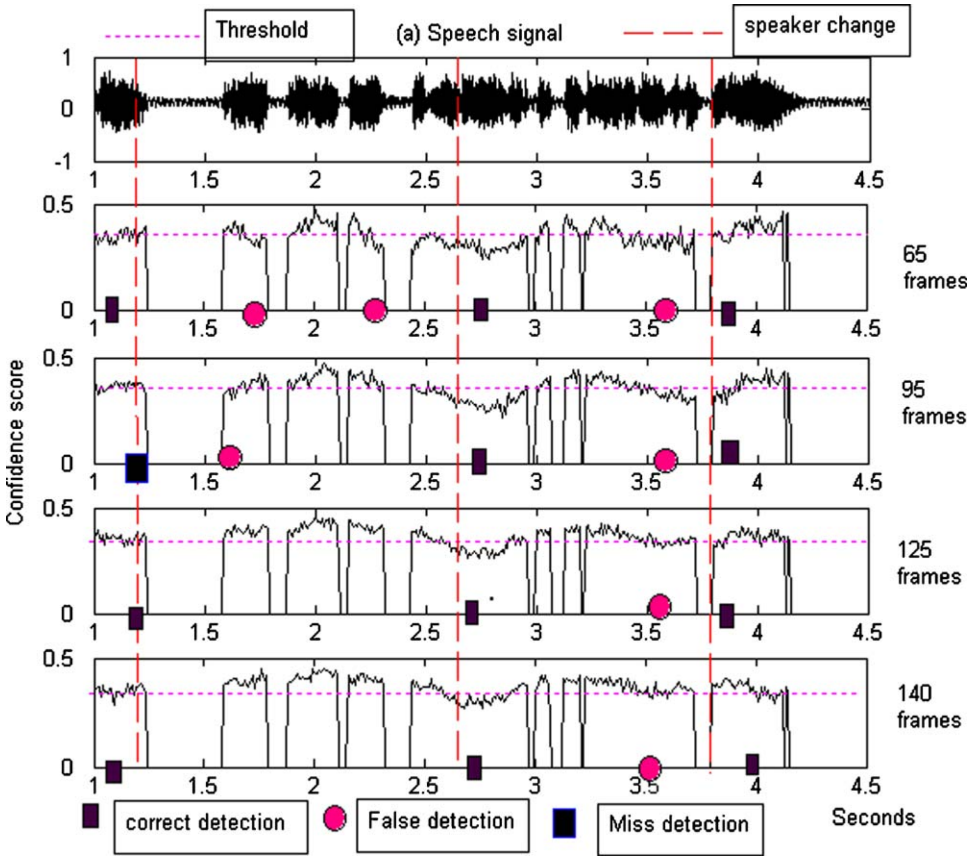

Fig. 7. Performance of the algorithm for various analysis window size.

frames is appropriate to detect speaker change for short duration speech segments.

Fig. 8 shows the evolution of the confidence score when the number of epochs increases. There is no significant change in the confidence score curve even though the number of epochs was increased to 1000. Hence the AANN models are trained for only 100 epochs.

It is not possible to obtain the same average confidence score for all the true speaker change points. The average confidence score of speaker change point will be low when compared to the average confidence scores of the frames on either sides of the speaker change point. So the local minima of the average confidence scores is considered instead of global minimum. To avoid the FA, the local minima which are less than the threshold value are considered. Hence, after obtaining the average confidence scores for the entire speech signal as described in Section 4.1, the hypothesized speaker change point is validated by using the threshold. Table 1 summarizes the determination of a for 125 frames of analysis window and 100 epochs. The parameter a is selected to achieve over segmentation. Table 1 shows that low miss detection is achieved for $a = 0.5$ and it is used in our experiments.

The segmented speech is then clustered as described in Section 4.2. The performance of the proposed method for speaker clustering for varying cluster parameter t_c is given in Table 2. From this table, it is of clear evidence that the best asp of 89.62% and the acp of 99.12% are obtained for $t_c = 0.2$.

Experimental results show that the clustering algorithm gives best performance when there is no segmentation errors. The performance of the diarization system is mostly depends on the performance of the segmentation algorithm.

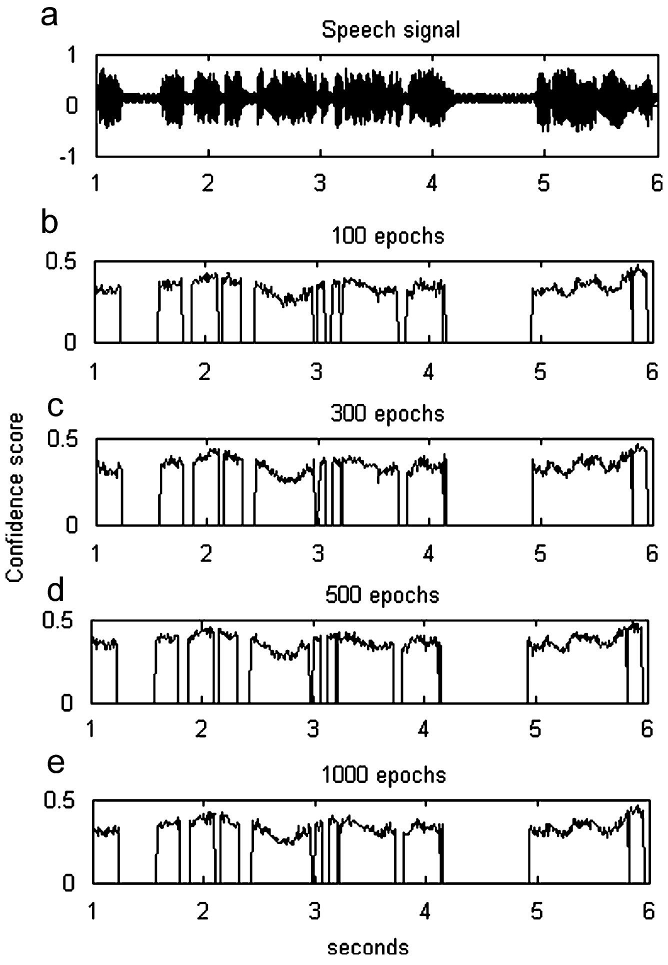

Fig. 8. Effect of epochs on the confidence score.

Table 1
Determination of adjustable parameter a.

Parameter a	0.25	0.3	0.35	0.4	0.45	0.5	0.55
α_r (%)	0	0	14.6	14.6	17.1	23	29.1
β_r (%)	25.2	25.2	17.3	17.3	7.6	0	0

Table 2
Determination of t_c.

Parameter t_c	0.15	0.17	0.19	0.2	0.21
Average speaker purity (%)	55.52	75.52	88.88	89.62	65.36
Average cluster purity (%)	97.16	98.28	99.12	99.12	62.73

Table 3
Error rates (MS,FA, SE and DER) obtained with our approach for the AUdata and RT’03S databases.

Corpus	MS (%)	FA (%)	SE (%)	DER (%)
AUdata	0.9	3.2	7.62	11.72
NIST RT’03S	0.92	3.28	7.81	12.01

6.4. Results

This section presents the results obtained on the AUdata and RT’03S databases. Based on the previous experiments the parameters selected are analysis window size = 125 frames, epochs = 100, $a = 0.5$ and $t_c = 0.2$. Table 3 shows the performance of the overall diarization system in terms of MS, FA, SE and DER. For AUdata, the DER is 11.72% whereas the DER for RT’03S is 12.01%.

This result has been compared with the results reported in Meignier et al. (2006), where the results obtained for NIST RT’03S database are given for two approaches namely CLIPS system and LIA system. The CLIPS system relies on a classical step by step strategy. It involves a GLR distance based detector strategy and followed by a hierarchical clustering. The LIA system is an integrated strategy which is based on an evolutive hidden Markov modeling (E-HMM) of the conversation. In this iterative approach, both the segmentation and the speaker models are used at each step and are re-evaluated at the next step. The best errors of these approaches are such that DER is between 12.9% and 24.7%, whereas the DER obtained from our method is 12.01%.

The results are also compared with the results reported in Fergani et al. (2008) where one class support vector machine (SVM) based dissimilarity measure has been used for speaker change detection and speaker clustering steps of the speaker diarization method and 12.28% of DER has been obtained for NIST RT’03S database. So the performance obtained from the proposed method competes with the methods reported in the literature.

7. Conclusion

In this paper we have presented an alternate method for speaker diarization using MFCC features and AANN. The proposed approach relies on a classical strategy based on speaker turn detection followed by a clustering process. The distribution capturing ability of the AANN model is utilized for segmenting the conversation and grouping each segment into one of the speaker classes. The experimental results show that the proposed approach competes with the standard speaker diarization methods reported in the literature and it is an alternative method to the existing speaker diarization methods.

This paper is mainly dedicated to speaker diarization experiments on broadcast news data. The future work will be in the direction to study the performance of the proposed algorithm for other domains such as recorded meetings and telephone conversations. The hardest task definitely corresponds to meeting data with very spontaneous speech, overlapping voices, disfluencies, distant speakers and back ground noise. So the future work will be devoted to a better adaptation of the acoustic features to the proposed approach and to some prior processing to be implemented before performing speaker diarization on recorded meetings.

References

Ajmera, J., Wooters, C., 2003. A robust speaker clustering algorithm. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 411–416.

Bianchini, M., Frasconi, P., Gori, M., 1995. Learning in multilayered networks used as autoassociators. IEEE Trans. Neural Networks 6, 512–515.

Bourlard, H., Kamp, Y., 1988. Auto association by multi layer perceptrons and singular value decomposition. Biol. Cybernet. 59, 291–294.

Chen, S.S., Gopalakrishnan, P., 1998. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 127–132.

Cheng, S., Wang, H., 2004. Metric SEQDAC: a hybrid approach for audio segmentation. In: Proceedings of the 8th International Conference on Spoken Language Process, pp. 1617–1620.

Davis, S.B., Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust. Speech Signal Process. 28, 357–366.

Delacourt, P., Wellekens, C., 2000. DISTBIC: a speaker based segmentation for audio data indexing. Speech Commun. 32, 111–126.

Fergani, B., Davy, M., Houacine, A., 2008. Speaker diarization using one-class support vector machines. Speech Commun. 50, 355–365.

Gauvain, J.L., Lamel, L., Adda, G., 1998. Partitioning and transcription of broadcast news data. In: International Conference on Spoken Language Processing, pp. 1335–1338.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs, NJ.

Kishore, S.P., 2000. Speaker verification using autoassociative neural networks model. M.S. Thesis, Department of Computer Science and Engineering, Indian Institute of Technology Madras.

Kotti, M., Moschou, V., Kotropoulas, C., 2007. Speaker segmentation and clustering. Signal Process. 88, 1091–1124.

Kramer, M.A., 1991. Nonlinear principal component analysis using auto associative neural networks. AIChE 37, 233–243.

Meigneir, S., Bonastre, J.F., Igounet, S., 2001. E-HMM approach for learning and adapting sound models for speaker indexing. In: Proceedings of the Speaker Odyssey—The Speaker Recognition Workshop, pp. 175–180.

Meignier, S., Moraru, D., Fredouille, C., Bonastre, J.F., Besacier, L., 2006. Step by step and integrated approaches in broadcast news speaker diarization. Comput. Speech Lang. 20, 303–330.

Moraru, D., Besacier, L., Castelli, E., 2004. Using a priori information for speaker diarization. In: Proceedings of the Odyssey 2004 Workshop on Speaker Recognition, pp. 355–362.

NIST, 2004. Fall 2004 Rich Transcription (RT-04F) ⟨www.nist.gov/speech/tests/rt/rt2004/fall/docs/ rto4feval-plan-v14.pdf⟩.

Palanivel, S., 2004. Person authentication using speech, face and visual speech. Ph.D. Thesis, Department of Computer Science and Engineering, Indian Institute of Technology Madras.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted Gaussian mixture models. Digital Signal Process. Rev. J. 10 (1–3), 19–41 special issue on NIST 1999 Speaker Recognition Workshop.

Sieglar, M., Jain, U., Raj, B., Stern, R., 1997. Automatic segmentation, classification and clustering of broadcast news audio. In: Proceedings of the DARPA Speech Recognition Workshop, pp. 97–99.

Sinha, R., Tranter, S.E., Gales, M.J.F., Woodland, P.C., 2005. The Cambridge University March 2005 speaker diarization system. In: Proceedings of the European Conference on Speech Communications and Technology, pp. 2437–2440.

Siu, M.H., Rohlicek, R., Gish, H., 1992. An unsupervised, sequential learning algorithm for segmentation of speech waveforms with multi speakers. In: Proc. of the IEEE International Conference on Acoustic, Speech, and Signal Processing, pp. 189–192.

Solomonoff, A., Mielke, A., Schmidt, M., Gish, H., 1998. Clustering speakers by their voices. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 757–760.

Tranter, S.E., Reynolds, D.A., 2006. An overview of automatic speaker diarization systems. IEEE Trans. Audio Speech Lang. Process. 14 (5), 1557–1565.

Wilcox, L., Kimber, D., Chen, F., 1994. Audio indexing using speaker identification. In: Proceedings of the SPIE Conference on Automatic Systems for the Inspection and Identification of Humans, pp. 149–157.

Yegnanarayana, B., 1999. Artificial Neural Networks. Prentice-Hall, New Delhi.

Yegnanarayana, B., Kishore, S.P., 2002. AANN: an alternative to GMM for pattern recognition. Neural Networks 15, 459–469.