Middle-East Journal of Scientific Research 24 (12): 3858-3868, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.3858.3868

Speaker diarization using Support Vector Machines and AutoAssociative Neural Network

[1]J. Gladson Maria Britto and [2]S. Suresh Kumar

[1]Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, India
[2]Principal, Vivekanandha college of Engineering for Women, Thiruchencode, India

Abstract: This paper proposes a new method for speaker diarization using support vector machines (SVM) and autoassociative neural network (AANN). The speaker diarization process consists of segmenting a conversation speech signal into homogeneous segments which are then clustered into speaker classes. The proposed method uses SVM and AANN models to capture the speaker specific information from Predictive Cepstral Coefficients (LPCC), Mel-Frequency Cepstral Coefficients (MFCC). The distribution capturing ability of the SVM and AANN models are utilized for segmenting the conversation speech and grouping each segment into one speaker classes. The proposed method have been tested on different databases the experimental results shows that the proposed approach competes with the standard speaker diarization methods and found that the AANN gives better performance than SVM.

Key words: Linear Predictive Coefficients (LPC) · Linear Predictive Cepstral Coefficients (LPCC) · Mel-Frequency Cepstral Coefficients (MFCC) · Weighted Linear Predictive Cepstral Coefficients (WLPCC) · Support Vector Machines (SVM) · Autoassociative Neural Network (AANN)

INTRODUCTION

Speaker diarization is the task of automatically partitioning a conversation speech signal involving multiple speakers into homogeneous segments and grouping together all the segments that correspond to the same speaker. The first part of the task is known as speaker turn point identification and speaker segmentation while the second one is called as speaker clustering hence speaker turn point identification followed by speaker clustering is known as speaker diarization [1-4].

Speaker turn point detection involves determining the points at which there is a speaker turn changes in the multi speaker speech data as in audio recordings of conversation, broadcast news and movie [5]. Speaker turn point identification is the first step in the speaker based segmentation of multi speaker only. Speaker segmentation is important for tasks such as audio indexing, speaker tracking and speaker adaptation in automatic transcription of conversational speech [6]. Speaker turn point identification should do without the knowledge of the number of speakers and the identity of speakers. Therefore, a Speaker turn point identification systems should be speaker independent.

The existing approaches for Speaker turn point identification are based on the dissimilarity in the distributions of data before and after the points of speaker change. Dissimilarity measurement is commonly based on comparison of the parametric statistic model of the distribution such as Mahalanobis distance, Weighted Euclidean distance, Bayesian information criteria [7-11]. In these approaches for Speaker turn point identification, the dissimilarity is measured for the data between two adjacent windows of fixed length [12]. The points at which the dissimilarity is above a threshold are hypothesized as the speaker turn points [13]. We propose an approach in which a classification model is trained to detect the Speaker turn points and segment the data for according to the speakers, i.e., speaker clustering.

The important contribution of this paper concerns the use of the distribution capturing ability the SVM and AANN for speaker turn point identification and speaker clustering for speaker diarization. The proposed method

Corresponding Author: J. Gladson Maria Britto, Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, India.

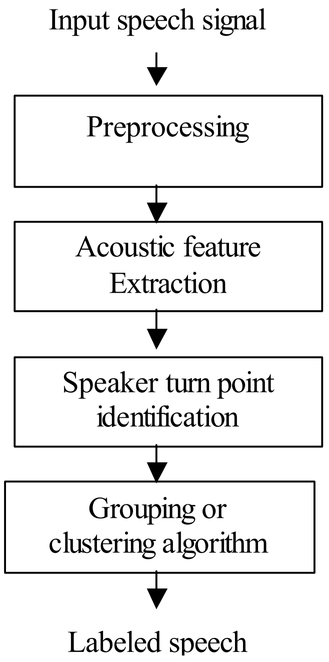
Input speech signal
Preprocessing
Acoustic feature Extraction
Speaker turn point identification
Grouping or clustering algorithm
Labeled speech

Fig. 1: Block diagram of speaker diarization

gives a classical two step speaker diarization approach based on speaker turn point identification followed by a speaker clustering process in Figure.1.

The next part of the paper is organized as follows a brief description about the method of extracting specific features from the speech signal is described in section 2. SVM and AANN model distribution of acoustic feature vectors are 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. Section 7 concludes the paper

Feature Extraction For Speaker Turn Point Identification And Speaker Segmentation: Acoustic features representing the speaker information can be extracted from the speech signal at the segmental features are the features extracted from the short (20 milliseconds) segments of the speech signal. These features represent the short time spectrum speech signal [14]. The short time spectrum envelop of the speech signal is attributed primarily to the shape of the same sound uttered by two persons may differ due to change in the shape of the individual's vocal tract system and the manner of speech production. For acoustic feature extraction, the differenced speech signal is divided in to frame of 20 milliseconds, with a shift of 10 milliseconds. Feature extraction is done by using LPC, LPCC and MFCC.

Data Collection: Two speaker Conversation speech signal is recorded.

- Male-male conversation.
- Male-Female conversation.
- Female-Female conversation.

The recording rate for speech is 8 KHz. The sampling bit rate is 16 bits. The recording mode is mono. The sample values vary from -32,768 to +32,767.We used unidirectional microphone. For 1 second 8000 samples will be recorded. Frame size is 20 milliseconds (160 samples).Frame shift is 10 milliseconds (80 samples).The file is stored as. wav extension format.

LPC Model: Linear Predictive Coefficients (LPC) is a powerful speech analysis technique. It is predominant technique for estimating the basic speech parameters. e.g. pitch, formants and vocal tract area function and for representing speech for low bit rate transmission or storage.

- Linear prediction coefficients.
- Linear prediction (LP) analysis.

Each sample is predicted as linear weighted sum of past p samples, where p is the order of LP analysis.

$$\text{x (n)} = \sum_{K=1}^{p} a_k\, x(n-k) \qquad (1)$$

The predicted signal value is given in the above equ. 1

x(n) is the predicted signal value.
x(n-k) is the previous observed values.
a_k is the predictor coefficients.
For example p = 14

$x(15) = a_1x(14)+a_2x(13)+\ldots+ a_{14}\,x$ (1)
$x(16)= a_1x\,(15)+a_2x(14)+\ldots+ a_{14}\,x$ (2)
….. ….. ……
$x(160)=a_1x(159)+a_2x(158)+..+a_{14}x\,(146)$ (2)

Linear prediction coefficients (LPC) $\{a_k\}$ are determined from the above equations.

The basic idea behind the LPC model is that given a speech sample at time n,s(n) can be approximated as linear combination of the past p speech samples.

The linear prediction analysis is to determine the set of predictor coefficient $\{a_k\}$, directly from the speech signal. So that the spectral properties of the digital from the below match those of the speech wave from with in the analysis window. Since the spectral characteristics of the speech vary over time, the predictor coefficients at a given time N must be estimated from a short segment of the speech signal occurring around time n. Thus the basic approach is to find a set of prediction coefficients that minimize the mean squared prediction error over a short segment of the speech wave form.

Durbin's recursive solution for the autocorrelation equation is used for finding LPC Coefficients.

Autocorrelation Method: A simple and straight forward way of defining the limits on m in this summation is to assume that the speech segment $S_n(m)$, is identically zero outside the intervals $0 \le m \le (n-1)$. Thus the speech sample for the minimization can be expressed as

$$S_n(m) = s(m+n).\ w(m),\ 0 \le m \le n-1 \qquad (3)$$
$$= 0 \text{ other wise}$$

Autocorrelation Analysis: Each frame of windowed signal is next autocorrelated to give

$$r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m) s_n(m+i-k) \qquad (4)$$

where,
r_l is the energy of the 1^{th} frame using equ.(3).
Where the highest autocorrelation value P is the order of LPC. P values from 8 to 20 are used.

R_l (0) = energy of the 1st frame.

In this paper, p=16 gives better performance compared to other order of LPC. The autocorrelation function is symmetric, i.e. $r_n(-k) = r_n(k)$, the LPC equations can be expressed as
Autocorrelation equation

$$\sum_{k=1}^{p} \hat{a}_k r_n(|i-k|) = r_n, 1 \le i \le p \qquad (5)$$

where, a_k is the predictor coefficients.

This is expressed in matrix form

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ \vdots \\ r_n(p) \end{bmatrix} \qquad (6)$$

Durbin's Recursive Solution for Autocorrelation Equation: The most efficient method for solving this particular system of equation is Durbin's recursive procedure which can be stated as. The process of solving for the predictor coefficients for the predictor of an order p. the solution for the predictor coefficient of all order less than P have also been obtained (i.e) the predictor coefficient for a predictor of order 2.

$$E^{(0)} = r\ (0)$$

$$k_i = \left\{ r(i) - \sum_{j=1}^{L-1} \alpha_j^{(i-1)}\ r(|i-j|) / E^{(i-1)},\ 1 \le i \le p \right\} \qquad (7)$$

$$\alpha_i^{(i)} = K_i$$

$$\alpha_j^{(1)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}$$

where,
LPC coefficients $= a_m = \alpha_m^{(p)}$, $1 \le m \le p$
K_m = PARCOR coefficients.

LPC Parameter Conversion to Linear Predictive Cepstral Coefficients (LPCC): A very important LPC parameter set, which can be derived directly form the LPC coefficient set, is the LPC cepstral coefficients, c(m). The recursion used is

$$C_0 = \ln\sigma^2 \qquad (8)$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m} \right) c_m a_{m-k}, 1 \le m \le p \qquad (9)$$

$$c_m = + \sum_{k=1}^{m-1} \left(\frac{k}{m} \right) c_m a_{m-k},\ m > p \qquad (10)$$

where,
σ^2 is the gain term in the LPC model.
C_0 to C_m are LPCC coefficients

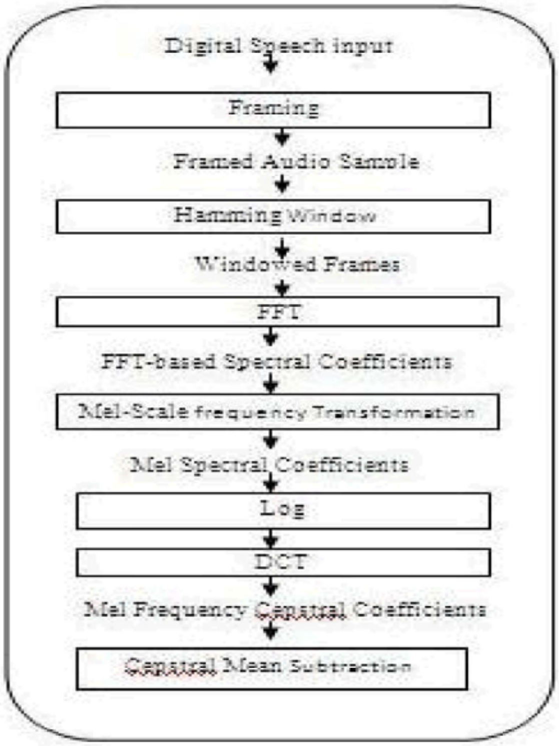
Digital Speech input
Framing
Framed Audio Sample
Hamming Window
Windowed Frames
FFT
FFT-based Spectral Coefficients
Mel-Scale frequency Transformation
Mel Spectral Coefficients
Log
DCT
Mel Frequency Cepstral Coefficients
Cepstral Mean Subtraction

Fig. 2: MFCC feature extraction

Suppose, p=19 means 19th LPCC extracted from conversation speech signal [15]. The cepstral coefficients, which are the coefficients of the Fourier transform representation of the log magnitude spectrum.

The cepstral coefficients are more robust, reliable feature set for speech recognition than the LPC coefficient.

Mel-Frequency Cepstral Coefficients (MFCCs): MFCCs have been widely used in the field of Speaker turn point identification system and are able to represent the dynamic features of a signal as they extract both linear and non-linear properties. The Mel-frequency Cepstral Coefficients (MFCC) is a type of wavelet in which frequency scales are placed on a linear scale for frequencies less than 1 kHz and on a log scale for frequencies above 1 kHz.

The complex cepstral coefficients obtained from this scale are called the MFCC. The MFCC contains both time and frequency information of the signal and this makes them useful for feature extraction. The following steps are involved in MFCC computations in Figure 2.

2.5.1) Transform input signal, *x(n)* from time domain to frequency domain by applying Fast Fourier Transform (FFT), using.

$$\mathrm{Y(m)} = \frac{1}{\mathrm{F}}\sum_{\mathrm{n=0}}^{\mathrm{F\text{-}1}} \mathrm{x\ (n)\ w\ (n)\ e}^{-\mathrm{j}\frac{2\mathrm{p}}{\mathrm{F}}\mathrm{nm}} \tag{11}$$

where F is the number of frames, $1 \le n \le F$-1 and *w(n)* is the Hamming window function given by:

$$w(n) = \beta 0.5 - 0.5\cos\frac{2\pi n}{F-1} \tag{12}$$

where 1 = n = F-1 and $_\beta$ is the normalization factor defined such that the root mean square of the window is unity.

2.5.2) Mel-frequency wrapping is performed by changing the frequency to the Mel using the following equation.

$$\mathrm{mel} = 2595 \text{ x } \log_{10}(1+\frac{f_{\mathrm{HZ}}}{700}) \tag{13}$$

Mel-frequency wrapping uses a filter bank, spaced uniformly on the Mel scale. The filter bank has a triangular band pass frequency response, whose spacing and magnitude are determined by a constant Mel-frequency interval.

2.5.3) The final step converts the logarithmic Mel spectrum back to the time domain. The result of this step is what is called the Mel-frequency Cepstral Coefficients. This conversion is achieved by taking the Discrete Cosine Transform of the spectrum as:

$$c_m^i = \sum_{n=0}^{\mathrm{F1}} \cos\ (\mathrm{m}\frac{\pi}{\mathrm{F}}(n+0.5)\ \log_{10}(H_n) \tag{14}$$

where $0 \le m \le L - 1$ and L is the number of MFCC extracted form the ith frame of the signal. H_n is the transfer function of the nth filter on the filter bank. These MFCC are then used as a representation of the signal.

SVM Modeling For Capturing The Distribution Of Acoustic Featre Vectors: A support vector machine (SVM) [16] is a machine learning technique that learns the decision surface through a process of discrimination and has good generalization characteristics. SVM is based on the principle of structural risk minimization. Like RBFNN (Radial Basis Function Neural Network), support vector machines can be used for pattern classification and non linear regression. Support vectors are used to find hyper plane between two classes. Support vectors are close to the hyper plane. Support vectors are the training samples that define that optimal separating hyper plane and are difficult patterns to classify. For linearly separable data, SVM finds a separating hyper plane, which separates the data with the largest 18 margins. For linearly separable data, it maps the data in the input space into high dimension space $x Є R^1 \rightarrow \Phi (x) Є R^H$ with kernel function Φ (x), to find the separating hyper plane. Each SVM separates a single class form all the remaining classes (one-vs.-rest approach).

Given a set of features corresponding to N subjects for training, N SVMs are created. Each SVM is trained to distinguish between features of a single subject and all other features in the training set. During testing, the distance from x to the SVM hyper plane is used to accept or reject the identity claim of the subject.

Inner product kernel maps input space to higher dimensional feature space. Inner product kernel $K (x, x_i) = \Phi (x).\Phi (x_i)$.

Where x is input patterns, x_i is support vectors.

For example,

assume $x = [x_1, x_2]^t$ is input patterns

$X_i = [x_{i1}, x_{i2}]^t$ is support vectors

$$
\begin{aligned}
K (x, x_i) &= (x^t x_i)^2 \\
&= (x_1 x_{i1} + x_2 x_{i2})^2 \\
&= x_1^2 x_{i1}^2 + x_2^2 x_{i2}^2 + 2\, x_1\, x_2\, x_{i1}\, x_{i2} \qquad (15)
\end{aligned}
$$

where,

$\Phi (x) = (x_1^2, x_2^2, 1.414\, x_1\, x_2)$

$\Phi (x_i) = (x_{i1}^2, x_{i2}^2, 1.414\, x_{i1}\, x_{i2})$

$K (x, x_i) = \Phi (x).\Phi (x_i)$

SVM maps two-dimensional input space to three-dimensional feature space that is shown in Figure. 3.

SVM Modeling for Speaker Turn Point Identification: The overlapping frame is calculated manually.

Each frame in LPCC (19th order) must ends with +1 or-1. +1 indicates the overlapping frame (Speaker turn point). -1 indicates the non-overlapping frame.

SVM Training: The manually calculated overlapping frames are appended with +1 and non-overlapping frames are appended with -1 using SVM Torch.

SVM Testing: Test conversation LPCC values are given to SVM test. The result is stored in the result.dat.

The result.dat file contains positive (+1) and negative (-1) values. Positive values indicate overlapping frames in the conversation file. That is, Speaker turn points.

AANN Modelling For Capturing the Distribution of Acoustic Feature Vectors: Autoassociative neural network 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 [17, 18].

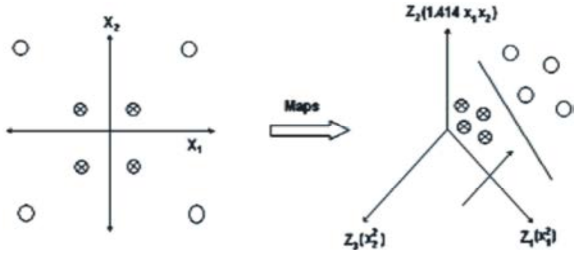

Fig. 3: SVM Maps 2-Dimensional input space to 3-Dimensional input Space

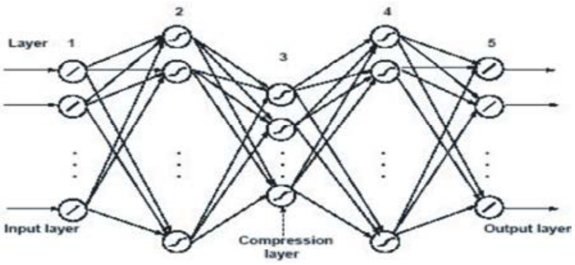

Fig. 4: A five layer AANN model

A five layer autoassociative neural network model, as shown in Figure 4, is used to capture the distribution of the feature vectors in our study. 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 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.

The structure of the AANN model used in our study is 19L 38N 5N 38N 19L, where L denotes a linear and N denotes a nonlinear units. The nonlinear output function for each unit is *tan h* (s), where s is the activation value of the unit. The standard back propagation learning algorithm is used to adjust the weights of the network to minimize the mean square error for each feature vector. 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 AANN captures the distribution of the input data

In order to visualize the distribution capturing ability, one can plot the error for each input data point in the form of some probability surface. The error ei for the data point i in the input space is plotted as pi= exp (-ei/α), where α is a constant. Note that *pi* is not strictly a probability density function, but we call the resulting surface as probability surface. The plot of the probability surface shows large amplitude for smaller error *ei* indicating better match of the network for that data point.

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

The Proposed Speaker Diarization Algorithm: The speaker diarization involves the technical elements presented in the previous section and two main steps: speaker turn point identification and speaker clustering is defined in jothi lakshmi *et al*. (2009).

The outline of the algorithm is shown in Figure. 5 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 turn point occurs at the middle frame (i.e.,) RHB and LHB will be from different speakers and all the feature vector from the RHB may not fall into the distribution and the model gives low confidence score. Likewise, if the middle frame is not the true speaker turn point and both LHB and RHB are from the same speaker then the confidence score of RHB will be 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 turn 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 [1]. After detecting the speaker changes, the segments obtained must be clustered to determine the number of speakers

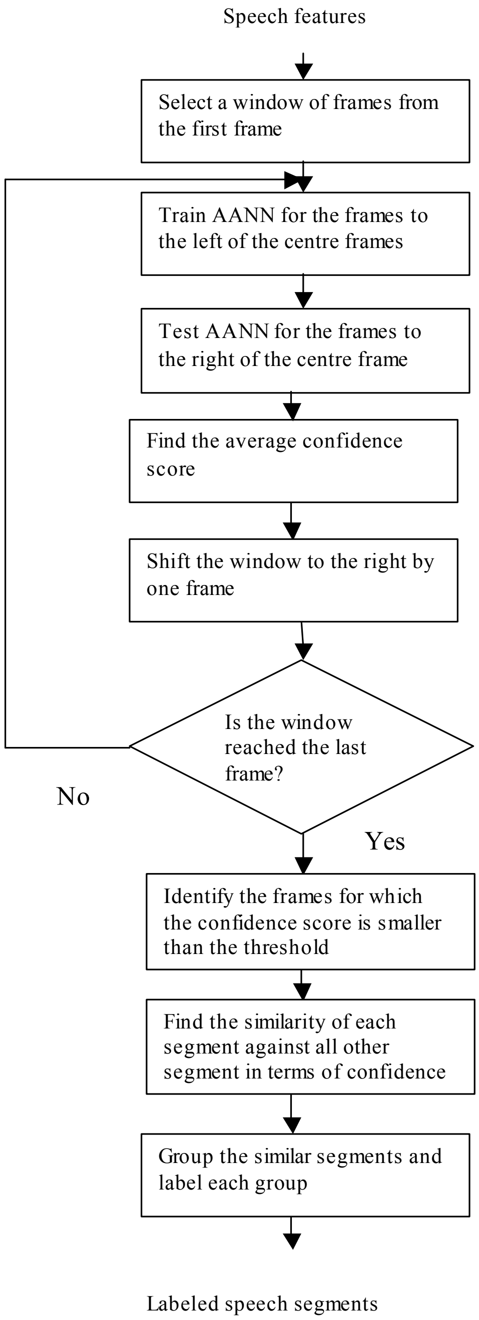
Speech features
Select a window of frames from the first frame
Train AANN for the frames to the left of the centre frames
Test AANN for the frames to the right of the centre frame
Find the average confidence score
Shift the window to the right by one frame
Is the window reached the last frame?
No
Yes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

Fig. 5: flow chart of the proposed speaker diarization system

Speaker Turn Point Identification: We begin with the assumption that there is a Speaker turn point located in the data stream at the centre 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 half of the window may not fall into the distribution of the feature vectors from the left half window [1]. On the contrary, if the speech signal of this analysis window comes from only one speaker then the feature vectors is in the right half of the window falls into the distribution of feature vectors of the left half window [7].

Given the speech feature vectors S = {si : i = 1,2,…., n} where i is the frame index and n is the total number of feature vectors in the speech signal;. The proposed

algorithm for detecting Speaker turn point is given below:
(1) From n frames, m number of feature vectors (m mod 2) = 1 are considered for k_{th} analysis window W_k and is given by

$$W_k=\{S_j\},\ k = j < m+k \qquad (16)$$

- It is assumed that the Speaker turn point occurs at the middle feature vector (c) of the analysis window.

$$c = k + \frac{m}{2} \qquad (17)$$

We consider all the feature vectors in the analysis window Wk that are located left of c as left half window (L_k).

$$L_k = \{S_j\},\ k = j< c\text{-}1 \qquad (18)$$

Similarly, all the feature vectors that are located right of c is in right half window (R_k).

$$R_k = \{s_j\},\ c+1 = j< m+k \qquad (19)$$

- AANN is trained using the feature vectors in L_k and the model captures the distribution of this block of vectors. Then feature vectors in R_k are given as input to eh AANN model and the output of model is compared with the input to compute the normalized squared error e_k. The normalized squared error (ek) for the feature vector y is given by

$$e_k = \frac{\|y - o\|^2}{\|y\|^2} \qquad (20)$$

where **o** is the output vector given by the model. The error e_k is transformed into a confidence score s using

$$S = exp(-e_k) \qquad (21)$$

If true Speaker turn point occurs at c, then L_k and R_k will be from different speakers and the confidence score s for this c will be low. Likewise, if c is not the true Speaker turn point and both L_k and R_k are form the same speaker then the confidence score s will be high. The next possibility is either L_k or R_k may have the speech feature vectors from both the speakers. If this is the case, the confidence score s will be in between the above two values.

- The value k is incremented by one and the steps from 1 to 4 are repeated until m+k reaches n.

It is not possible to obtain the same confidence score for all true Speaker turn point. The confidence score of Speaker turn point will be low when compared to the confidence scores of the frames on either side of the Speaker turn point. So the local minimum of the confidence score us considered instead of global minimum. To avoid the false alarms, the local minima which are less than the threshold value are considered. Hence, after obtaining the confidence score for the entire speech signal the hypothesized Speaker turn point is validated by using a threshold. The threshold (t) is calculated from the confidence score as

$$T = s_{\ min} + as_{\ min} \qquad (22)$$

where s_{min} is the global minimum confidence score and a is the adjustable parameter. The proposed method is unsupervised because it can detect the speaker changes without any knowledge of the identity of speakers and there is no need for training speaker models beforehand.

The thresholding step is performed as in any other detection algorithm: the threshold is tuned in accordance to some tradeoff between false alarms (FA) and missed detections. 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 near to 0 the number of miss detections will be more. If it is near to 1 the number of FA will be more. So the parameter is a 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.

Speaker Clustering: Once the speaker turn points have been detected, the next important step is speaker clustering. It contains of labeling segments of speech, detected by the speaker turn 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 member of segments obtained from the segmentation algorithm. The proposed algoritm for speaker clustering is described as follows:

- For each segment ci, the AANN is trained using the frames in ci and the model captures the distribution of this block of data. Then the feature vector of each segment cj is given as input to the AANN model and the output of the model is compared with the input to compute the confidence score sij which is the confidence score of j^{th} trained segment. From the outcome of this step, a confidence score matrix s_{mat} of size P xP can be formed such that

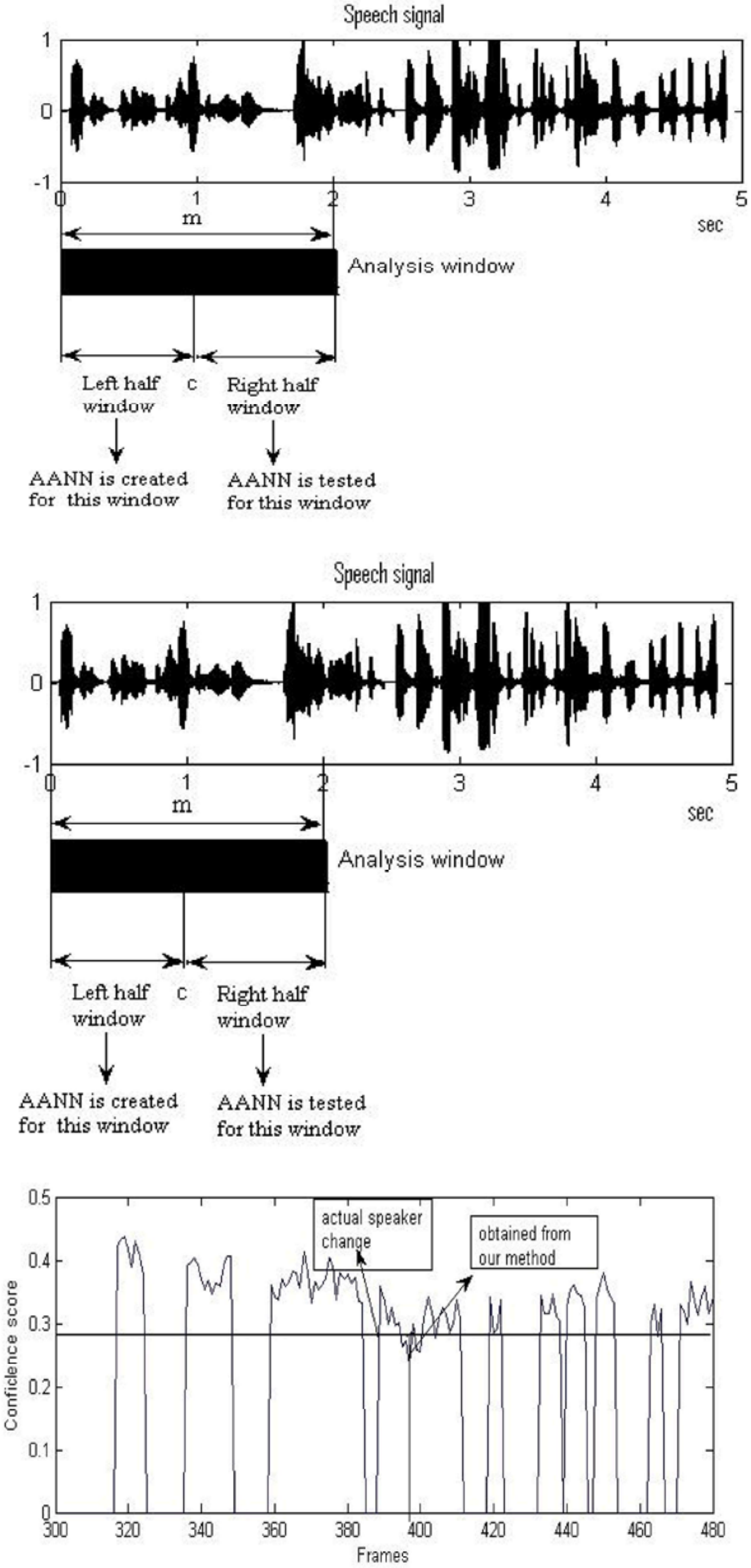

Fig. 6: Concept of the proposed segmentation algorithm (a) Speech signal (b) confidence score

$$S_{mat} = \begin{pmatrix} s_{11} & s_{12} \cdots s_{1p} \\ s_{21} & s_{22} \cdots s_{2p} \\ & \end{pmatrix}$$

- 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

$$S_{ik} = \begin{cases} (s_{ij} + s_{ji})/2 & if\ k \geq i \\ 0 & if\ k < i \end{cases}$$

- The distance between ith segment and all other segments are computed by

$d_{ik} = s_{ij} - s_{ik}$ $k>i$

now s_{mat} becomes

- If

$D_{ik}<s_{ij}t_c$, $0<t_c<0.5$, $0<i=P$, $i<k=P$

ith segment and kth segment are most 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 and k thsegment 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.

Performance Measures: The performance of speaker segmentation is assessed in terms of two types of error related to speaker turn point identification. A false alaram (FA) (α) of speaker turn occurs when a detected speaker turn point is not a true one. A missed detection (β) occurs when a true speaker turn point cannot be detected. The FA rate (α) and missed detection rate(β) are defined as [2, 19].

$$\alpha = \frac{\text{Number of false alarms}}{\text{Number of actual speaker changes} + \text{Number of false alarms}} \quad (24)$$

$$\beta = \frac{\text{Number of missed detections}}{\text{Number of actual speaker changes}} \quad (24)$$

Experimental Results: This section presents experimental results using different speech databases.

For our experiments on Speaker turn point identification, we use the extended data consist of two-speaker conversation. A total dataset of 9 conversations is used in our studies. This dataset includes three conversations for each of male-male, male-female and female-female speaker conversations. The Speaker turn point in the conversation is manually marked. The total dataset is divided in to training dataset a validation dataset and test dataset.

The speech data is processed using a frame size of 20 milliseconds. Each frame size is represented by a19 dimensional LPCC feature vector and MFCC feature vector. The speech data of the conversations in the training dataset is processed to obtain positive and negative examples for training the Speaker turn point identification using SVM and AANN. The speech data of conversations in the validation dataset is used for obtaining the negative examples to train the false alarm reduction SVM. The speech data of the conversation in the test dataset is used for evaluating the performance of the Speaker turn point identification system.

The speech data of conversation is given as the input to the Speaker turn point identification system. The sliding window method is used to obtain hypotheses from the Speaker turn point identification SVM. The output of the SVM is smoothed to eliminate the short duration speaker turns. The hypotheses after removal of short speaker turns are processed by the false alarm reduction. The Speaker turn point identification performance is measured as the missed detection rate (MDR) and the false alarm rate (FAR). The missed detection rate is defined as the ratio of the number of Speaker turn points missed (M) and the number of actual Speaker turn point (A) are given in equation (25 and (26).

$$MDR = \frac{M}{A} X\ 100 \quad (25)$$

$$FAR = \frac{F}{T - A} X\ 100 \quad (26)$$

where F is the number of false hypotheses and T is the number of test patterns.

The MDR and FAR are determined at different stages of the Speaker turn point identification system.

The performance of the different window length is given in the Table 1.

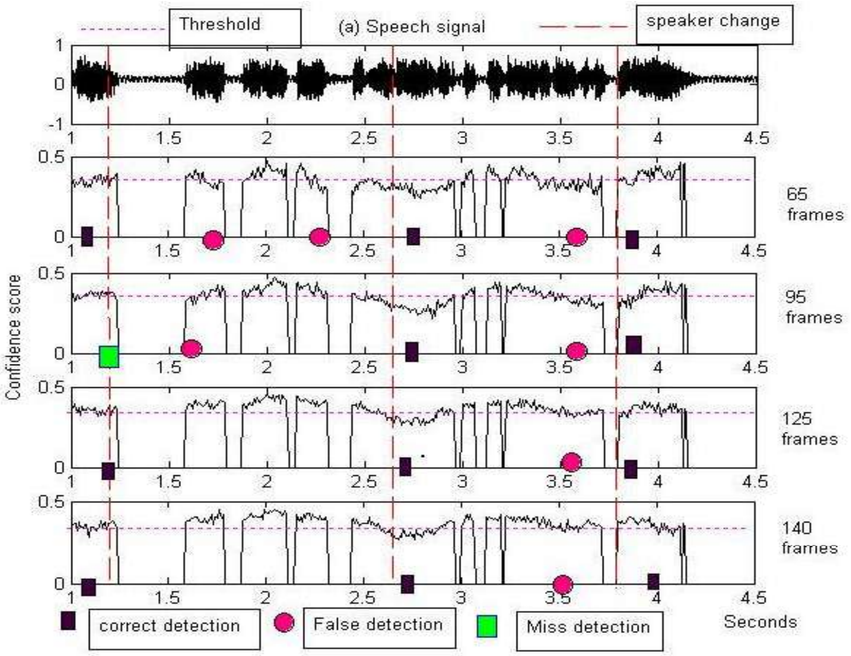

Fig. 7: Shows the performance for the various analysis window sizes

Table 1: Performance of the Speaker turn point detection system at various stages.

Window size(frames)	Missed Detection Rate (MDR)
5	5.13
10	8.99
15	11.23
20	3.24

Table 2: Performance of the Speaker turn point identification system comparison (LPCC and MFCC).

Classifier	α_r	β_r
AANN	15.75%	4.63%
SVM	27.01%	25.10%

The Figure 7. shows the performance for the various analysis window sizes.

After the identification of the speaker turn points the segmented speech is clustered as described in section 4.2

Table 2 shows the performance of the speaker turn point identification system comparison (LPCC and MFCC) using SVM and AANN.

CONCLUSION

In this paper we have presented a alternate method for speaker diarization using LPCC and MFCC features with SVM and AANN. The proposed approach relies on a classical strategy based on speaker turn point identification followed by speaker clustering process the distribution capturing ability of the SVM and AANN model was utilized for segmentation. The speech and grouping each segment into one of the speaker classes. The proposed method is an alternative method to the existing speaker diarization method, the future work will be in the direction to study the performance of the proposed algorithm for other domain such as recorded meetings and telephone conversations. So the future work will be dedicated to a better adaptation of the acoustic feature to the proposed approach and to some prior processing to be implemented before performing speaker diarization on recorded meetings.

REFERENCES

1. Jothilakshmi, S., S. Palanivel and V. Ramalingam, 2010. Unsupervised Speaker Segmentation using Autoassociative Neural Network, 2010 International Journal of Computer Applications, (0975-8887) 1(7).
2. Cheng, S. and H. Wang, 2004. Metric SEQDAC: a hybrid approach for audio segmentation. In Proceedings of the 8th International Conference on Spoken Language Processing, pp: 1617-1620.
3. Po-Chuan Lin, Jia-Chingwiang, Jhing-Fa Wang and Hao-Ching Sung, 2007. Unsupervised Speaker turn point detection using SVM Training Misclassification Rate, IEEE Transactions on Computers, 56(9).

4. Shilei zhang shuwu zhang and Bo Xu, 2006. A Two-Level Method For Unsupervisied Speaker-based Audio Segmentation. The 18th International Conference on pattern Recognition (ICPR'06) IEEE, 2006.
5. Kim, H., D. Elter and T. Sikora, 2005. Hybrid speaker based segmentation system using model level clustering. In Proceedings of the IEEE International conference on Acoust. Speech, Signal Processing (ICASSP 05), pp: 745-748.
6. Jitendra Ajmera, Iain Mccowan and Herve Bourlard, 2004. Robust Speakers change Detection. IEEE Signal Processing Letters, 2(8).
7. Yegnanarayana, B. and S.P. Kishore, 2002. AANN: An alternative to GMM for pattern recognition. Neural Networks, 15: 459-469.
8. Noureddine ELLOUZE, 2006. Robustness Improvement Of Speaker Segmentation Techniques Based on the Bayesian Information Criterion, IEEE, 2006.
9. Andre G. Adami, Sachin S. Kajarekar and Hynek Hermansky, 2002. A New Speaker turn point Detection Method For Two-Speaker Segmentation" IEEE, 2002.
10. Guillaume Lathoud Iain A. McCowan, 2003. Location Based Speaker Segmentation", IEEE, 2003.
11. Amit S. Maleganonkar, Aladdin M. Ariyaeeinia and Perasiriyn Sivakumaran, 2007. Efficient Speaker turn point detection using adapted Gaussian mixture models, IEEE Transactions on Audio, Speech and Language Processing, 15(6).
12. Mergnier.s Moraru, D., C. Fredouline, J.F. Bonastre and L. Beasier, 2006. step by step and integrated approaches in broadcast news speaker diarization, comput. Speech Lang, 20: 303-330.
13. Margarita Kotti, Emmanouil Benetos and Jaime S. Cardoso, 2006. Automatic Speaker Segmentation Using Multiple Features And Distance Measures: Comparison of Three Approaches, 1-4244-0367-7/06 IEEE, 2006.
14. Ajmera, J., I. McCowan and H. Bourland, 2004. Robust speaker change detection. IEEE Signal Process. Lett., 11(8): 649-651.
15. Malegaonkar, A., A. Ariyaeeinia, P. Siva Kumaran and J. Fortuna, 2006. Unsupervised Speaker turn point Detection Using Probabilistic Pattern Matching, IEEE SIGNAL PROCESSING, LETTERS, 13(8).
16. Trainter, S.E. and D.A. Reynolds, 2006. An overview of automatic speaker diarization systems. IEEE Trans. Audio Speech Lang. process, 14(5): 1557-1565.
17. Vapnik, V., 1998. Statistical learning theory, John Wiley and Sons, New York.
18. Kartik, V., D. Srikrishna Sathish and C. Chandra Sekar, 2006. Speaker turn point detection using Support Vector Mechines, "Speech and Vision Laboratory, Indian Institute of Technology Madras, pp: 1-5.
19. Delacourt, P. and C.J. Wellekens, 2000. DISTBIC: a speaker based segmentation for audio data indexing. Speech Comm., 32: 111-126.