ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Speaker Segmentation using Support Vector Machines and Auto Associative Neural Network

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

[1]*Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, India.*
[2]*Principal, Vivekananda college of Technology for Women, Thiruchencode, India.*

ARTICLE INFO

Article history:
Received 3 October 2015
Accepted 31 October 2015

Keywords:
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).

ABSTRACT

In this paper we propose a classification based method to identify speaker turn point detection and segmenting speech contains individual speaker using support vector machines (SVM) and Auto associative neural network (AANN). Speaker turn point detection is important for automatic segmentation of multi speaker speech data into homogenous segments with each segment containing the data of one speaker only. Existing approach for speaker turn point detection are based on the dissimilarities of the distribution of data before and after a speaker turn point. Patterns extracted from the data around the speaker turn points are used as positive examples. Patterns extracted from the data between the speaker turn point are used as negative examples. The linear predictive cepstral coefficients (LPCC) and Mel frequency cepstral coefficients (MFCC) and extracted from the speech signal, the positive and negative examples are used in training a SVM and AANN separately for speaker turn point detection. The extraction of fixed length pattern from speaker are given as input to SVM and AANN models are used to classify the speaker turn points and speaker no turn points using specific features. Experiments are carried out on different audio databases and the proposed method is better for detecting speaker turn point changes with sort duration of speech.

To Cite This Article: J.Gladson Maria Britto and Dr. S. Suresh Kumar., Speaker Segmentation using Support Vector Machines and Auto Associative Neural Network ***Aust. J. Basic & Appl. Sci.,*** *9(33): 37-44, 2015*

INTRODUCTION

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. Speaker turn point detection 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. Speaker turn point detection should do without the knowledge of the number of speakers and the identity of speakers. Therefore, a Speaker turn point detection systems should be speaker independent.

The existing approaches for Speaker turn point detection 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. In these approaches for Speaker turn point detection, the dissimilarity is measured for the data between two adjacent windows of fixed length. The points at which the dissimilarity is above a threshold are hypothesized as the speaker turn points. We propose an approach in which a classification model is trained to detect the Speaker turn point points and segment the data for according to the speakers.

1. Acoustic feature extraction:

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. 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 and LPCC.

Corresponding Author: J.Gladson Maria Britto , Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, India.
E-mail: gmbritto@gmail.com

Australian Journal of Basic and Applied Sciences, 9(33) October 2015, Pages: 37-44

***1. a) Data Collection*:**

Two speaker Conversation speech signal is recorded.

i. Male-male conversation.
ii. Male-Female conversation.
iii. 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.

***1. b) 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.

i. Linear prediction coefficients.
ii. Linear prediction (LP) analysis.

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

$$x(n) = \sum_{k=1}^{p} a_k x(n-k) \quad (1)$$

The predicted signal value is given in the above Equation (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

$$x(15) = a_1 x(14) + a_2 x(13) + ... a_{14} x(1)$$
$$x(16) = a_1 x(15) + a_2 x(14) + ... a_{14} x(2)$$
$$x(160) = a_1 x(159) + a_2 x(158) + ... a_{14} x(146) \quad (2)$$

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.

***1.c) 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 $0 \le m \le (n-1)$

$$S_n(m) = s(m+n).w(n), \quad 0 \le m \le (n-1) \quad (2a)$$

= 0 other wise

Autocorrelation analysis:

Each frame of windowed signal is next autocorrelated to give

$$r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} S_n(m) S_n(m+i-k) \quad (3)$$

where,

r_1 is the energy of the l[st] frame using Equation (3).

Where the highest autocorrelation value P is the order of LPC. P values from 8 to 20 are used.

$R_1(0)$ = energy of the 1[st] 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

$$\sum_{k=1}^{p} \hat{a}_k r_n(|i-k|) = r_n \quad ,1 \le i \le p \quad (4)$$

where a_k is the predictor coefficients.

This is expressed in matrix form

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \dots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \dots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \dots & r_n(p-3) \\ \vdots & \vdots & \vdots & \vdots & \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \dots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ \\ r_n(p) \end{bmatrix} \quad (5)$$

***1. d) 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.

$$E^{(0)} = r(0)$$
$$k_i = \left\{ r(i) - \sum_{j=1}^{L=1} \alpha_j^{(i-1)} r(|i-j|) \Big/ E^{(i-1)} \quad ,1 \le i \le p \right\} \quad (6)$$
$$\alpha_i^{(i)} = k_i$$
$$\alpha_j^{(1)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$$

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

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 from the LPC coefficient set, is the LPC cepstral coefficients, c (m). The recursion used is

$$C_0 = \ln \sigma^2 \tag{7}$$

$$c_m = a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_m a_{m-k,} 1 \le m \le p \tag{8}$$

$$c_m = + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_m a_{m-k,} \text{ m > p,} \tag{9}$$

where,

σ^2 is the gain term in the LPC model.

C_0 to c_m are LPCC coefficients

Suppose, p=19 means 19th LPCC extracted from conversation speech signal. 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.

***1. e) Mel-frequency Cepstral Coefficients (MFCCs)*:**

MFCCs have been widely used in the field of Speaker turn point detection system and are able to represent the dynamic features of a signal as they extract both linear and non-linear properties. MFCC can be a useful tool of feature extraction in vibration signals as vibrations contain both linear and non-linear features. 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 contain both time and frequency information of the signal and this makes them useful for feature extraction. The following steps are involved in MFCC computations.

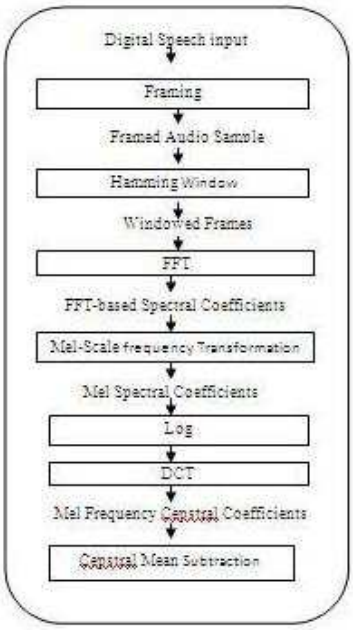

Fig. 1: MFCC feature extraction

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

$$Y(m) = \frac{1}{F}\sum_{n=0}^{F-1} x(n)w(n)e^{-j\frac{2\pi}{F}nm} \tag{10}$$

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

$$w(n) = \beta\, 0.5\cos\frac{2\pi n}{F-1} \tag{11}$$

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

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

Australian Journal of Basic and Applied Sciences, 9(33) October 2015, Pages: 37-44

$$\text{mel} = 2595 \text{ x } \log_{10}\left(1 + \frac{f_{HZ}}{700}\right) \quad (12)$$

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.

c) 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

$$c_m^i = \sum_{n=0}^{F1} \cos\left(\mathrm{m}\frac{\pi}{\mathrm{F}}(n+0.5)\log_{10}(H_n)\right) \quad (13)$$

where $0 \leq m \leq L-1$ and L is the number of MFCC extracted from the i[th] frame of the signal. H_n is the transfer function of the n[th] filter on the filter bank. These MFCC are then used as a representation of the signal.

2. *Modeling for Speaker turn point detection:*

A support vector machine (SVM) 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 → Φ (x) Є R^H with kernel function

Φ (x), to find the separating hyper plane. SVM was originally developed for two class classification problems. The N class classification problem can be solved using NSVMs. 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

$K(x, x_i) = (x^t x_i)^2$

$= (x_1 x_{i1} + x_2 x_{i2})^2$

$= x_1^2 x_{i1}^2 + x_2^2 x_{i2}^2 + 2\, x_1 x_2 x_{i1} x_{i2}$ (14)

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 Fig.2

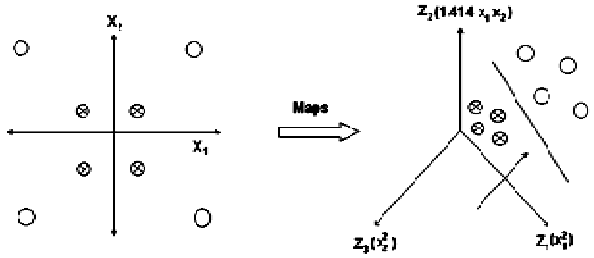

Fig. 2: SVM Maps 2-Dimensional input space to 3-Dimensional input Space.

2. a) *SVM principle*:

It is one of the popular classification models. Support vectors are used to find hyper plane between two classes. Maps input space to higher dimensional feature space. Support vectors are the training samples that define the optimal separating hyper plane. Support vectors are close to the hyper plane.

2. b) *SVM modeling for Speaker turn point detection*:

The overlapping frame is calculated manually:

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

3. AANN model 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.

A five layer autoassociative neural network model, as shown in Figure 3, 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 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 tanh (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.

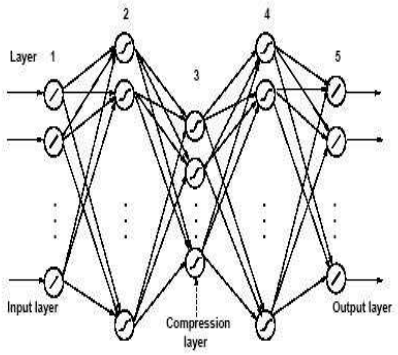

Fig. 3: A five layer AANN model

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 turn point detection algorithm:

This paper purposes a novel Speaker turn point detection algorithms using AANN. The basic concept of the proposed method is illustrated in Figure 2. 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. 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.

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:m number of feature vectors (m mod 2) = 1 are considered for kth analysis window W_k and is given by

$$W_k = \{Sj\}, k \le j < m + k \quad (15)$$

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

$$c = k + \frac{m}{2} \quad (16)$$

Australian Journal of Basic and Applied Sciences, 9(33) October 2015, Pages: 37-44

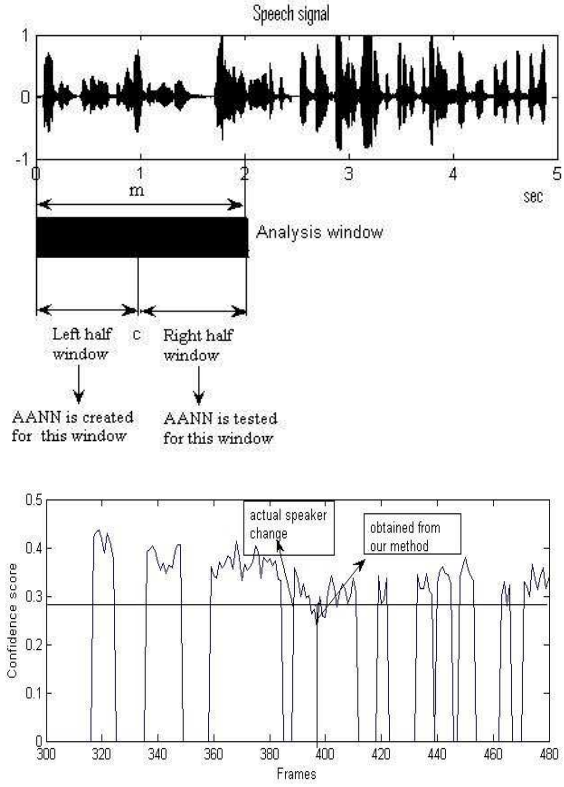

Fig. 4 : Basic concept of the proposed algorithm.

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

$$Lk = \{Sj\}, k \le j \le c-1 \quad (17)$$

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

$$Rk = \{Sj\}, c+1 \le j \le m+k \quad (18)$$

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

$$e_k = \frac{\|y-0\|^2}{\|y\|^2} \quad (19)$$

where 0 is the output vector given by the model. The error e_k is transformed into a confidence score S using

$$S = \exp(e_k) \quad (20)$$

If true Speaker turn point occurs at c, then Lk and Rk 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 point and both Lk and Rk are form the same speaker then the confidence score s will be high. The next possibility is either Lk or Rk 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 points. The confidence score of Speaker turn point point will be low when compared to the confidence scores of the frames on either side of the Speaker turn point 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

$$T = s\min + as\min \quad (21)$$

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.

Australian Journal of Basic and Applied Sciences, 9(33) October 2015, Pages: 37-44

4. Experimental Results:

For our experiments on Speaker turn point detection, 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 detection 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 detection system.

The speech data of conversation is given as the input to the Speaker turn point detection system. The sliding window method is used to obtain hypotheses from the Speaker turn point detection 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 SVM to give the Speaker turn point detection. The Speaker turn point detection 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 point missed (M) and the number of actual Speaker turn point points (A) are given in equation (22) and (23).

$$MDR = \frac{M}{A} \times 100 \quad (22)$$

$$FAR = \frac{F}{T - A} \times 100 \quad (23)$$

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 detection system.

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

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

The performance of speaker segmentation is assessed in terms of two types of error related to Speaker turn point detections namely false alarms and missed detections. A false alarm (α) of Speaker turn point detection 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 false alarm rate (α_r) and detection rate (β_r) are defined as [16], [17].

$$\alpha_r = \frac{\text{Number of false alarmed speaker changes}}{\text{Number of detected speaker changes}} \quad (24)$$

$$\beta_r = \frac{\text{Number of missed detection}}{\text{Number of true speaker changes}} \quad (25)$$

Two other measures namely precision (p) and recall (r)can also be used, which are closely related to α_r , β_r [14], [15]. They are defined as

$$p = \frac{Number\ of\ correctly\ found\ spea\ker changes}{Total\ number\ of changes\ found} \quad (26)$$

$$r = \frac{Number\ of\ correctly\ found\ spea\ker changes}{number\ of\ actual\ spea\ker changes} \quad (27)$$

In order to compare the performance of different systems, the f-measure is often used and is given by

$$f = 2\frac{pr}{p+r} \quad (28)$$

The f – measure caries from 0 to 1, with a higher f – measure indicating better performance.

Table 2 shows Performance of the Speaker turn point detection system comparison (LPCC and MFCC). Using SVM and AANN.

5. Conclusion:

In this paper we have proposed an alternative method for speaker segmentation using LPCC and MFCC features and SVM and AANN.

The proposed algorithm can achieve effective speaker segmentation with data collection and it is capable of detecting speaker segments of shorten duration the algorithm can be applied for real time application and it does not require any prior knowledge about the speaker identity and their model. The proposed method was carried out for two speaker conversations and by using clean speech signals. This method can be used for speaker diarization

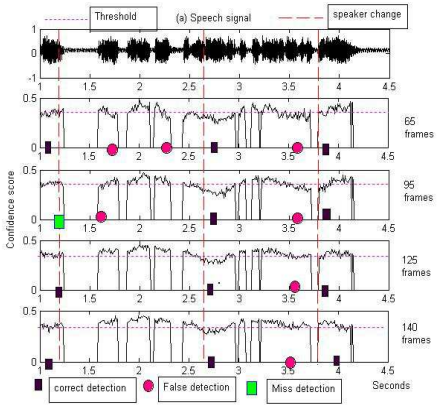

Fig. 5: Performance of the algorithm for various analysis window sizes.

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

Classifier	α_r	β_r	f-measure
AANN	15.77%	4.65%	89.03%
SVM	27.01%	25.11%	70.78%

REFERENCES

Jothilakshmi, S., S. Palanivel, V. Ramalingam, 2010. "Unsupervised Speaker Segmentation using Autoassociative Neural Network", International Journal of Computer Applications (0975 – 8887), 1-7.

Shilei zhang shuwu zhang, Bo Xu, 2006. "A Two-Level Method For Unsupervisied Speaker-based Audio Segmentation". The 18th International Conference on pattern Recognition(ICPR'06) IEEE.

Malegaonkar, A., A. Ariyaeeinia, P. SivaKumaran and J. Fortuna, 2006. "Unsupervised Speaker turn point Detection Using Probabilistic Pattern Matching" IEEE SIGNAL PROCESSING, LETTERS, 13(8).

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.

Jitendra Ajmera,Iain Mccowan and Herve Bourlard, "Robust Speakers change Detection." IEEE SIGNAL PROCESSING LETTERS, VOL.2, NO.8, AUGUEST 2004.

Andre G.Adami,Sachin S.Kajarekar, Hynek Hermansky, "A New Speaker turn point Detection Method For Two-Speaker Segmentation" IEEE, 2002.

Po-Chuan Lin, Jia-Chingwiang, Jhing-Fa Wang and Hao-Ching Sung. "Unsupervised Speaker turn point detection using SVM Training Misclassification Rate" IEEE TRANSACTIONS ON COMPUTERS, VOL.56, No.9, Sep 2007.

Noureddine ELLOUZE. "Robustness Improvement Of Speaker Segmentation Techniques Based on the Bayesian Information Criterion", IEEE, 2006.

Kartik, V. and 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, 1-5.

Guillaume Lathoud Iain A. McCowan, 2003. "LOCATION BASED SPEAKER SEGMENTATION", IEEE.

Margarita Kotti, Emmanouil Benetos, Jaime S. Cardoso, 2006. "Automatic Speaker Segmentation Using Multiple Features And Distance Measures: Comparison Of Three Approaches"1-4244-0367-7/06 IEEE.

Vapnik, V., 1998. "Statistical learning theory", John Wiley and Sons, New York.

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

Kim, H., D. Elter, 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), 745–748.

Ajmera, J., I. McCowan, H. Bourland, 2004. "Robust speaker change detection". IEEE Signal Process. Lett., 11(8): 649–651.

Delacourt, P., C.J. Wellekens, 2000. DISTBIC: "A speaker based segmentation for audio data indexing". Speech comm., 32: 111–126.

Cheng, S., H. Wang, 2004. Metric SEQDAC: "A hybrid approach for audio segmentation". In Proceedings of the 8th International conference on spoken language processing 1617– 1620.