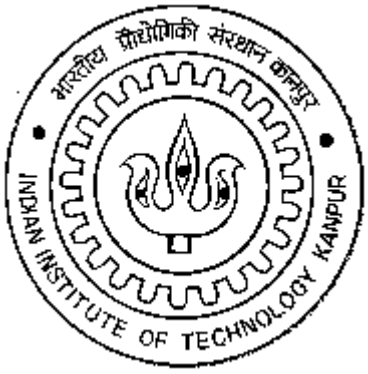
EE 492 – Project II

Audio Diarization

Project Guide: Dr. S.Umesh



Department of Electrical Engineering

IIT Kanpur

By:

Sanjeet Kumar (Y4380)

Sankalp Gulati (Y4383)

Acknowledgement

We are immensely grateful to Dr. S.Umesh, Associate Professor, Department of Electrical Engineering, IIT Kanpur for his valuable help, support and guidance in this project. He was instrumental in fostering interest in our project. Working with him and benefiting from his experience in the field was a very valuable learning experience for us.

Contents

1 Abstract …… 3

2 Introduction …….. 3

3 Diarization system framework ……. 5

3.1 Speech Detection……

3.2 Sex Classification…….....

3.3 Change Detection…….

3.4 Clustering…….

4 Implementation …….

4.1 Audio Classifier ……

4.2 Gender Classifier ……..

4.3 Change Detector ……

4.4 Clustering ……..

5 Conclusion …….

6 References …….

1. Abstract:

Audio diarization is the task of automatically segmenting an input audio stream into acoustically homogeneous segments and attributing them to sources. These sources can include particular speakers, music, background noise sources and other signal source/channel characteristics. Diarization has the utility in making automatic transcripts more readable and in searching and indexing audio archives.

2. Introduction:

Reduced cost of data storing devices and increased network bandwidth has allowed storage of large volumes of audio, including broadcasts, voice mails, meetings and other such documents. Thus there is a growing need for techniques which allow efficient and effective searching, indexing and accessing of these information sources. In addition to the fundamental technology of speech recognition, to extract the words being spoken, other technologies are needed to extract meta-data that provide context and information beyond the words.

Audio Diarization [1] or the marking and categorizing of audio sources within a spoken document, is one such technology. These audio sources may be the speakers in an audio file, so this technique would allow searching for words spoken by a speaker selectively in the categorized data. Other sources (non-speech) may be present in the audio file and so Diarization can also help in segmenting out commercials, deciding the structure of the broadcast program or in speech recognition systems for skipping the non speech sections which results in faster processing of the data.

Taking broadcast news as an example [1] which typically consists of speech from different speakers as well as music segments, commercials and noise (Figure 1). The types and details of the audio sources are application specific which can vary from a wide band source to a narrow band source. At the simplest diarization is speech versus nonspeech, where non-speech is a general class consisting of music, silence, noise, etc., that need not be broken out by type. A step ahead would be further marking where speaker changes occur in the detected speech and clustering segments of speech (a segment is a section of speech bounded by non-speech or speaker change points) coming from the same speaker. There may be applications where it may be desired to have more or less detail in the annotation of speech and non-speech classes (e.g., explicitly locate music, detect the narrow-band speech, label speech only by sex of the speaker etc.).

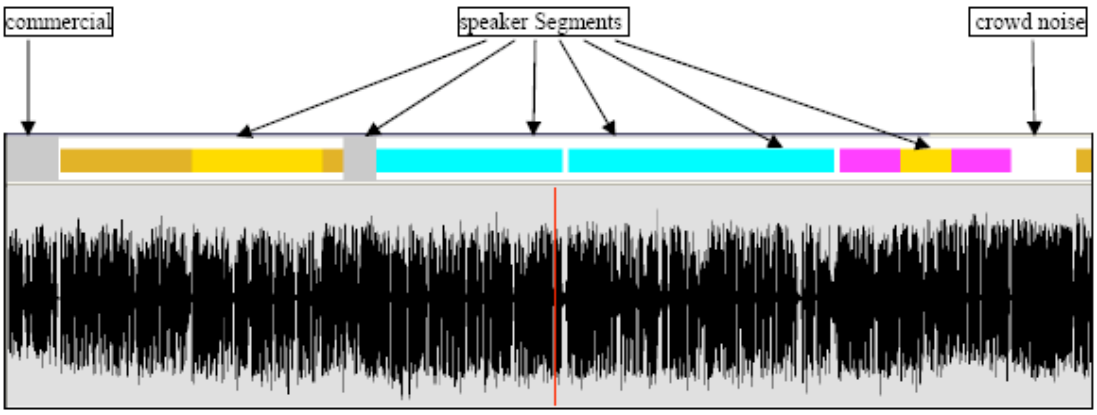


Figure 1: Broadcast news example of audio Diarization [1]

An advanced diarization system is expected to operate without any specific prior knowledge of speakers or the number of speakers in the audio. In the next section we present the general framework of diarization systems.

3 Diarization System Framework:

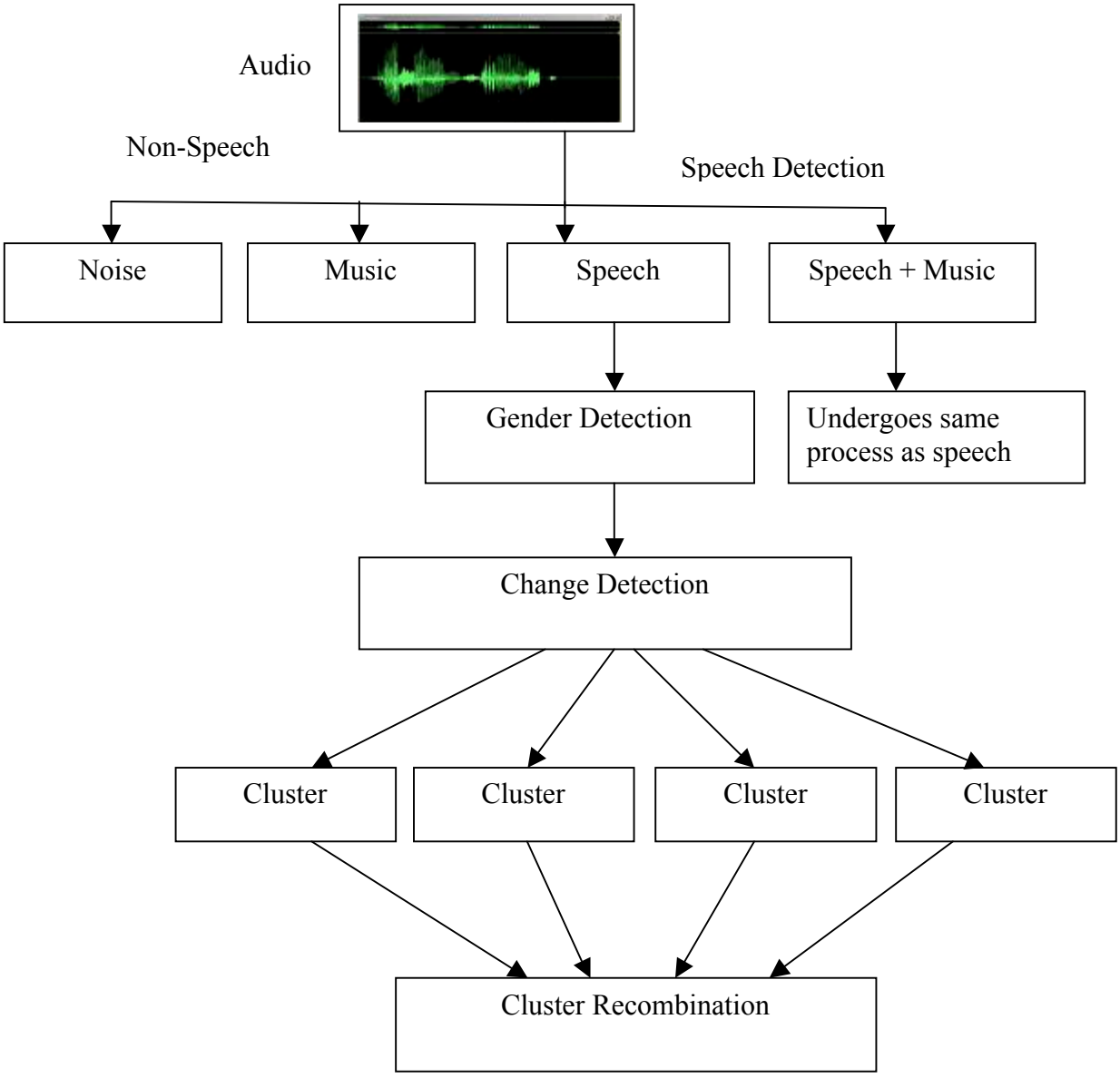


Figure 2: Basic Framework of Diarization

3.1 Speech Detection: Detecting regions of speech

The aim of this step is to find the regions of speech in the audio stream. Depending on the domain data being used, non-speech regions to be discarded can consist of many acoustic phenomena such as silence, music, room noise, background noise or cross-talk. The general approach used is maximum likelihood classification with Gaussian Mixture Models (GMMs) trained on labeled training data.

We have made models for speech, music and speech + music. The extra speech + xx models are used to help minimize the false rejection of speech occurring in the presence of music. Minimum length constraints and heuristic smoothing rules may also be applied.

Reviewing relevant prior work reveals that the most popular approach for audio segmentation is based on modeling the likelihood of a frame given a certain class with a Gaussian Mixture Model (GMM) using Mel Frequency Cepstral Coefficients (MFCC) features.

Let X be an entity we want to classify such as a speech segment for audio classification, an audio file for speaker recognition, a text for topic classification, etc. We assume X is represented by a sequence of feature vectors of length $n(x)$: $X=\{x1,\ldots,xn(X)\}$. The training set T is defined as a set of entities labeled with their class identity: $T = \{Xi, Ci\}$. The goal is given a test entity $Y=\{y1,\ldots,yn(Y)\}$, to classify it.

A common approach for computing the Maximum-Likelihood (ML) class for a test entity is assuming that the feature vectors are independent given a class identity Cj (equation 1):

$$\Pr(y_1, \ldots\ldots, y_{n(Y)} \mid C_j) = \prod_{i=1}^{n(Y)} \Pr(y_i \mid C_j) \qquad (1)$$

Speech cannot be assumed to be stationary during a long segment. Therefore, we choose to apply acoustic modeling on the frame level.

For GMM Training: We estimate a GMM for each class (speech, music, speech+music) using the feature vectors as training data. The acoustic feature vectors consist of 12 MFCCs, normalized log energy and the first and second differential coefficients of these. The test data is also divided into frames of length 20 ms with an overlap of 50% and feature vectors are extracted. Using these feature vectors, for each frame, a probability is calculated for every model and the frame is labeled with the model that gives the highest probability. Thus different segments of a given test data are labeled as speech, music or speech+music.

3.2 Gender Classification:

The aim of this step is to further classify the speech and speech+music segments according to the gender of the speaker. Classification for gender is typically done using maximum likelihood classification with GMMs trained on labeled training data. Two classifiers are run for speech (male/ female) and for speech+music (male+music/ female+music) each.

3.3 Change Detection: Detecting the boundaries of speaker change

The aim of this step is to find points in the audio stream likely to be change points between audio sources. If the input to this stage is the un-segmented audio stream, then the change detection looks for both speaker and speech/nonspeech change points. If a speech detector or gender classifier has been run first, then the change detector looks for speaker change points within each speech segment.

Two main approaches have been used for change detection. They both involve looking at adjacent windows of data and calculating a distance metric between the two, then deciding whether the windows originate from the same or a different source. The differences between them lie in the choice of distance metric and thresholding decisions.

The first general approach used for change detection, used is a variation on the Bayesian Information Criterion (BIC) technique. This technique searches for change points within a window using a penalized likelihood ratio test of whether the data in the window is better modeled by a single distribution (no change point) or two different distributions (change point). If a change is found, the window is reset to the change point and the search restarted. If no change point is found, the window is increased and the search is redone. [5]

A second technique uses fixed length windows and represents each window by a Gaussian and the distance between them by the Gaussian Divergence (symmetric KL-2 distance). [4] The peaks in the distance function are then found and define the change points if their absolute value exceeds a pre-determined threshold chosen on development data. Smoothing the distance distribution or eliminating the smaller of neighboring peaks within a certain minimum duration prevents the system over generating change points at true boundaries. Single Gaussians are generally preferred to GMMs due to the simplified distance calculations. Typical window sizes are 1-2 or 2-5 seconds when using a diagonal or full covariance Gaussian respectively. As with BIC the window length constrains the detection of short turns.

Since the change point detection often only provides an initial base segmentation for diarization systems, which will be clustered and often resegmented later, being able to

run the change point detection very fast is often more important than any performance degradation.

3.4 Clustering:

The purpose of this stage is to associate or cluster segments from the same speaker together. The clustering ideally produces one cluster for each speaker in the audio with all segments from a given speaker in a single cluster. The predominant approach used in diarization systems is hierarchical, agglomerative clustering with a BIC based stopping criterion consisting of the following steps:

0) Initializing leaf clusters of tree with speech segments.

1) Compute pair-wise distances between each cluster.

2) Merge closest clusters.

3) Update distances of remaining clusters to new cluster.

4) Iterate steps 1-3 until stopping criterion is met.

The clusters are generally represented by a single full covariance Gaussian but GMMs have also been used sometimes being built using mean-only MAP adaptation of a GMM of the entire test file to each cluster for increased robustness. The standard distance metric between clusters is the generalized likelihood ratio. It is possible to use other representations or distance metrics, but these have been found the most successful within the BIC clustering framework. The stopping criterion compares the BIC statistic from the two clusters being considered, x and y, with that of the parent cluster, z, should they be merged, the formulation being for the full covariance Gaussian case:

BIC $= LL - \frac{1}{2}\lambda P \log(N_z)$; $\Delta BIC = \frac{1}{2}\left(N_z \log|\Sigma| - N_1 \log|\Sigma_1| - N_2 \log|\Sigma_2|\right) - \lambda P$;

P = $\left(\frac{1}{2}d + \frac{1}{4}d(d+3)\right)\log(N_z)$ where M is the number of free parameters, N the number of frames, S the covariance matrix and d the dimension of the feature vector. If the pair of clusters is best described by a single full covariance Gaussian, the ΔBIC will be low, whereas if there are two separate distributions, implying two speakers, the ΔBIC will be high. For each step, the pair of clusters with the lowest ΔBIC is merged and the statistics are recalculated. The process is generally stopped when the lowest ΔBIC is greater than a specified threshold, usually 0. The use of the number of frames in the parent cluster, Nz, in the penalty factor, P, represents a 'local' BIC decision i.e. just considering the clusters being combined.

4. Implementation

The part of the project dealing with GMM modeling has been built using the HTK toolkit platform. HTK is the "Hidden Markov Model Toolkit" developed by the Cambridge University Engg Department. This toolkit aims at building and manipulating Hidden Markov Models (HMMs). The change detection algorithm has been implemented on MATLAB®.

4.1 Audio Classifier (Speech Detection)

The *Audio classifier* (first step towards diarization) for marking and categorizing of audio sources within a spoken document has been successfully implemented through

programming on HTK. For each of the sources - speech, music, speech+music a class model has been made using the GMM modeling of the feature vectors obtained by training of the speech data and extraction of MFCC feature vectors from the training data. The steps involved in building an audio classifier are as follows:

4.1.1 MFCC features extraction

Creating Mel Frequency Cepstral coefficients is a way of extracting the relevant information from an audio signal. They are the transformed version of a cepstral representation of an audio data (spectrum of a spectrum). The Mel scale is based on an empirical study of the human perceived pitch or frequency. The scale is divided into the units "Mel's". The Mel scale, generally speaking is a linear mapping below 1000 Hz and logarithmically spaced above. The mapping is usually done using the approximation:

$$F\_mel = 2595 * \log10 (1 + f/700) \qquad (4.1)$$

where F_mel is the perceived frequency in Mel.

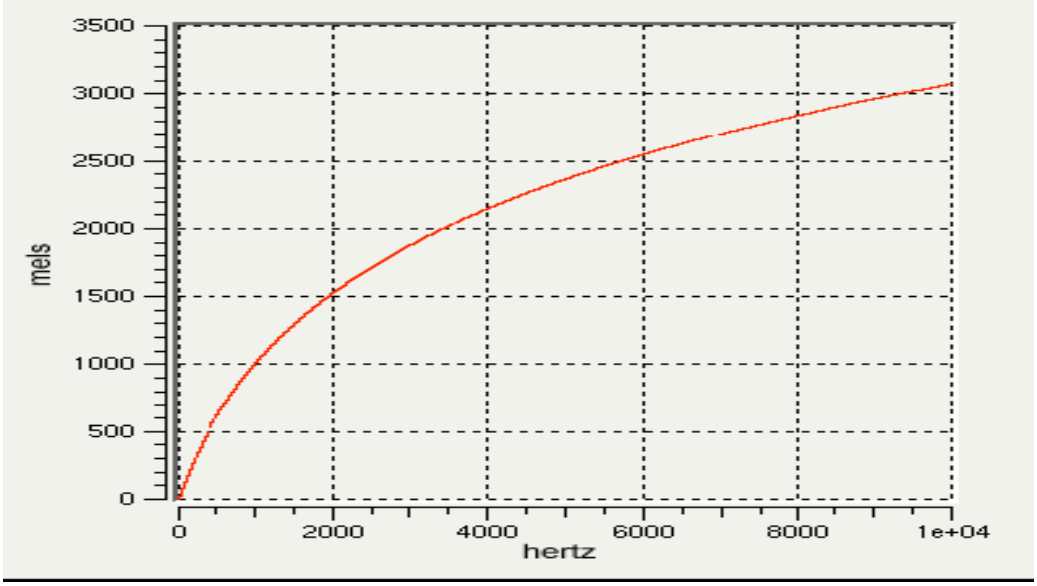


Figure 3: Mel transform (mapping frequency from linear to logarithmic scale)

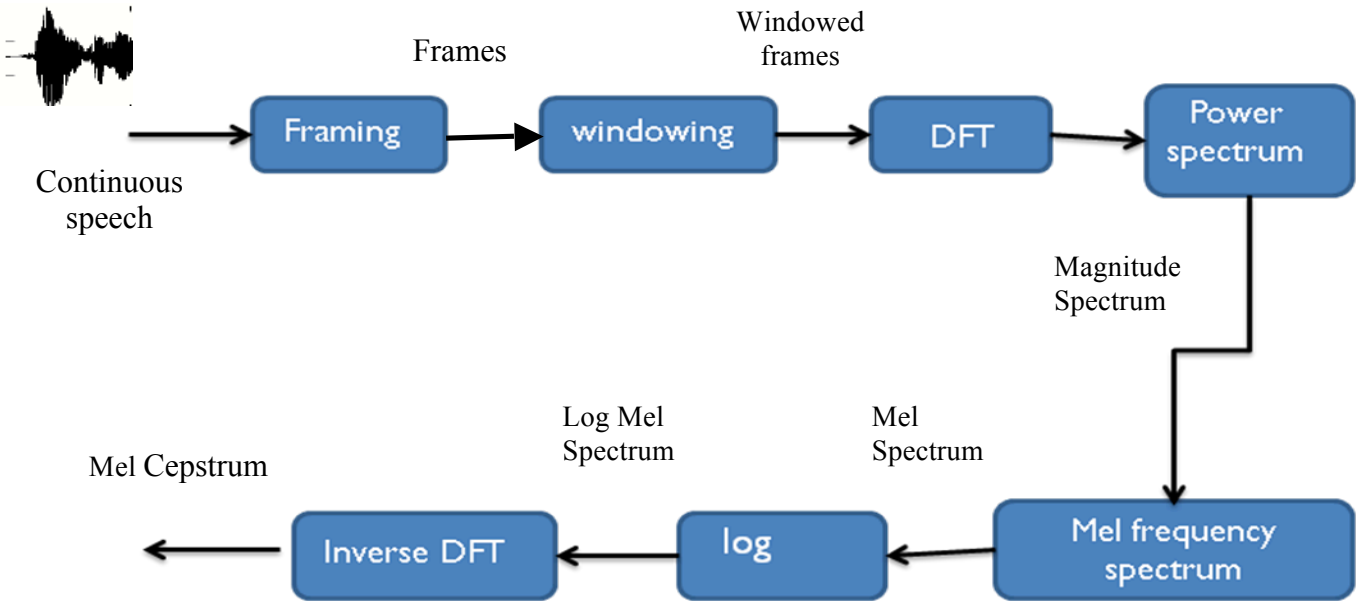


Figure 4: Steps involved in MFCC features extraction

The Mel cepstrum measures provide a good model of the speech signal. This is particularly true in quasi steady state voiced region of speech. This is the reason why frames of length 10-20 ms are chosen for extracting MFCC feature vectors as the speech signals are quasi steady in such small durations.

The delta (first derivative) and acceleration (second derivative) coefficients along with the energy measures are also used for the complete characterization of the speech signal.

4.1.2 GMM modeling using HTK

The MFCC feature vectors obtained by the analysis of the speech data are processed to obtain GMM models for each class of data (speech, music, speech +music, silence). The GMM modeling is being done in HTK using its inbuilt commands. HTK can only be used to build HMM model and so to build a GMM model we have chosen a HMM model with

three states (effectively one state, as the first and the last states are non-emitting i.e. no observation function).

The 512 mixture components GMM model of dimension D=39 is being made whose effective probability density function is given by:

$$b_j(o_t) = \sum_{k=1}^{M} c_{jk} b_{jk}(o_t) \qquad (4.2)$$

where, j=1 (as we are building GMM), M=512, O_t is the observation vector, C_{jk} is the weight of k^{th} component and $b_{jk}()$ is the D dimensional Gaussian density function of the k^{th} component with mean I_{jk} and covariance matrix $\sum_{jk}$.

The Gaussian density function can be found as:

$$b_{jk}(o_t) = \mathrm{N}(o_t, \mu_{jk}, \Sigma_{jk}) = \frac{1}{(2\Pi)^{D/2} \left|\Sigma_{jk}\right|^{1/2}} e^{\frac{-1}{2}(O_t - \mu_{jk})^T \Sigma_{jk}^{-1} (O_t - \mu_{jk})} \qquad (4.3)$$

4.1.3 Training Data

The set of training data being used for the modeling of different classes is taken from the CNN's broadcast news and is of one hour duration including speeches, music and several advertisements. The file has been divided into homogenous segments containing only one type of data belonging to a class. This task was performed manually over the whole file. The final duration of the training data are as follows:

S.No.	Class	Duration(min:sec)
1	Male	32:50
2	Female	11:03
3	Music + male	07:22
4	Music + female	02:11
5	Music	03:04

Table 1: Duration of the training data being used

4.1.4 Gender Classifier

The approach remains same as in speech detection part with the only difference being that the models that are used are male, female, male + music, female + music. Here also we are using 512 mixture GMM models.

4.1.5 Change Detector

For the detection of speaker change boundaries in a speech file we have used the KL2 (Kullback Leibler) distance metric also know as Gaussian divergence [4].

Relative Cross Entropy, or the Kullback Leibler (*KL*) distance between two Random Variables *A* and *B* is an information theoretic measure equal to the additional bit rate accrued by encoding random variable *B* with a code that was designed for optimal encoding of *A*. The larger this value, the greater the distance between to PDFs of the two Random Variables. It is formulated in the following way:

$$KL(A;B) = E_A\langle \log(P_A) - \log(P_B)\rangle \qquad (4.4)$$

Where, E_A is the expectation operation performed with respect to the PDF of A.

Since this expression is not symmetric, it is not strictly a distance metric. We therefore define the *KL2* metric as:

$$KL2(A;B) = KL(A;B) + KL(B;A) \tag{4.5}$$

When both A and B have Gaussian distributions we obtain:

$$KL2(A;B) = \frac{\sigma_A^2}{\sigma_B^2} + \frac{\sigma_B^2}{\sigma_A^2} + (\mu_A - \mu_B)^2\left(\frac{1}{\sigma_A^2} + \frac{1}{\sigma_B^2}\right) \tag{4.6}$$

The greater this value is, the greater is the distance between the two PDFs.

When A and B are vectors [6] (which is true in our case) the distance metric becomes:

$$KL2(A;B) = \Delta\mu_{AB}^T\left(\Sigma_A^{-1} + \Sigma_B^{-1}\right)\Delta\mu_{AB} + tr\left(\Sigma_A\Sigma_B^{-1} + \Sigma_B\Sigma_A^{-1} - 2I\right) \tag{4.7}$$

The implementation of this metric is being done in the following way:

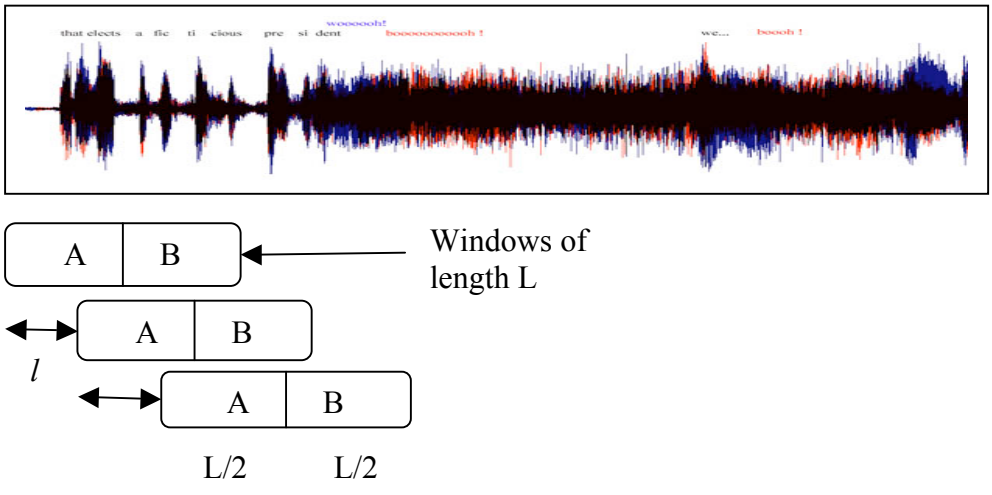


Figure 5: Scheme used for change detection

The sliding windows of length L were divided into two equal parts (A and B) , the GMM model for these two parts was made and then the KL2 distance metric between the two PDFs was calculated. The next window is then formed by shifting the window before by length *l*. Thus at each step we have this distance metric which is then plotted against the step index. When this KL2 distance reaches local maxima, speaker change was detected and a new segment boundary was formed. For the detection of peaks in the graph heuristics approach was used. Also some smoothening is being performed to reduce the percentage of false alarm (detecting a change which is not present) taking into account some assumptions like a speaker change will take a minimum of few seconds.

4.1.6 Testing of Speech Detector/Gender Classifier

The testing involves the Maximum likelihood characterization of the audio data at frame level. Test file generally contains multiple audio sources. Each frame is labeled as one of the classes (speech, music, speech +music, silence)/ (male, female, male + music, female + music). The result is a MLF file corresponding to the test vector, specifying the labels estimated.

4.1.7 Testing of the change detector

The automatically generated segment boundaries were compared with the hand-generated segments (which are the actual change speaker change points). The number of change points correctly detected is thus calculated and the ratio of this to the total number of change points present gives the accuracy of our change detector. And the ratio of correctly detected points to the total points detected gives the efficiency of the detector.

5. Results

The results of the change detector are summarized in the following table:

Definition used :

Efficiency = number of correctly detected points/ total detected points

Accuracy = number of correctly detected points/ total change points

S.No	Window length (L sec)	Number of mixtures	Increment(*l sec*)	Efficiency(%)	Accuracy(%)
1	2	1	0.25	16	85
2	2	3	0.25	13.3	87.5
3	2	1	1	16.5	95
4	5	1	1	36.7	82.5
5	5	3	1	31.7	80

6. Conclusions

Increase in window length will increase efficiency as number of points detected will reduce because the chance of detecting a point due to modulation decreases (as window length is large).

The other observation is that the increase in number of mixtures in GMM modeling tends to decrease the efficiency because modulation changes of a single speaker also get detected as we model finer details.

As the results above show, there is a sharp trade off between the accuracy and the efficiency. But our final result will not be affected by the lack of good efficiency because all the segments which correspond to the same speaker will be clustered together. The only price we have to pay is the increase in the computation of the next step which is to cluster the segments. On the other hand lack in accuracy will drastically degrade our final result because there will be two speakers present in a single segment and while modeling they will be modeled under a single name. Hence, we can afford to lose efficiency but cannot compromise on accuracy.

7. References

[1] D. A. Reynolds and P. Torres-Carrasquillo, "Approaches and applications of audio diarization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. V, Philadelphia, PA, Mar. 2005, pp. 953–956.

[2] Woodland P.C., Hain T., Johnson S.E., Niesler T.R., Tuerk A., Whittaker E.W.D & Young S.J. (1998), the 1997 HTK Broadcast News Transcription System. *Proc. DARPA Broadcast News Transcription System and Understanding Workshop*, pp. 41-48, Lansdowne, Virginia.

[3] Segmental modeling for audio segmentation Hagai Aronowitz, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598, U.S.A, haronow@us.ibm.com

[4] Automatic Segmentation, Classification and Clustering of Broadcast News Audio *Matthew A. Siegler, Uday Jain, Bhiksha Raj, Richard M. Ster,* ECE Department - Speech Group Carnegie Mellon University, Pittsburgh, PA 15213 msiegler@cs.cmu.edu uj@cs.cmu.edu bhiksha@cs.cmu.edu rms@cs.cmu.edu

[5] Speaker Environment and channel change detection clustering via the Bayesian Information Criterion, Scott Shaobing Chen & P.S. Gopalakrishnan, IBM T.J. Watson Research Centre, schen@watson.ibm.com

[6] Speaker Clustering using Direct Maximization of the MLLR-Adapted Likelihood, S.E. Johnson, P.C.Woodland, Cambridge University Engg. Department, Trumpington Street, Cambridge CB2 1PZ, U.K. {sej, pcw}@eng.cam.ac.uk