TECHNICAL UNIVERSITY OF CRETE

DEPARTMENT OF ELECTRONIC AND COMPUTER ENGINEERING

DIPLOMA DISSERTATION

# SPEAKER SEGMENTATION AND CLUSTERING ON GREEK BROADCAST NEWS

BY

FRAGKIADAKIS GEORGIOS

COMMITTEE:

DIGALAKIS VASSILIS, PROFESSOR

(SUPERVISOR)

KARYSTINOS GEORGIOS, ASSISTANT PROFESSOR

POTAMIANOS ALEXANDROS, ASSISTANT PROFESSOR

CHANIA

JUNE 2012

# SEGMENTATION AND CLUSTERING SPEAKERS ON GREEK BROADCAST NEWS

## CONTENTS

# LIST OF TABLES

# List of Images-Figures

# Abstract

Segmenting a broadcast signal that consists of numerous speakers has always posed a challenge to speech recognition systems. In this thesis we examine current knowledge, evaluate procedures implemented to date, identify obstacles and propose possible solutions.

Specifically, this thesis aims to identify and overcome the difficulties in the identification, classification, clustering and by extrapolation, the extraction of particular segments of speech. We focused on being able henceforth to detect segments which belong to different categories (e.g. music, speech, prerecorded advertisements, etc.) emphasizing on speaker turns. Initially, existing methods of speech classification and segmentation are enumerated from the international literature. Subsequently the Bayesian Information Criterion (BIC) was used to examine, evaluate and finally to experiment with those characteristics that would serve to clearly differentiate features of speech. The novel aspect of the thesis is the design and successful implementation of an updated BIC module that now enables one to delineate the non-homogenous nature of the signals. The new version of the BIC uses different pdfs to model a combination of characteristics. In the closing chapters, the adaptation formulae used to build the new system utilizing the transformation matrices produced by MLLR are discussed, before presenting an overall evaluation and comprehensive system that can be used to most successfully isolate and extract sections of speech at will.

# Chapter 1

## Introduction

### 1.1 Audio Segmentation Motivation and Applications

The recent rapid advances in technology have resulted in an ever-increasing potential of producing and processing huge amounts of digital information. Specifically, the quantities of recorded sound, speaking or audio, available on several databases on the Internet today is immense and therefore the successful management and access to such large amounts of data requires more efficient search engines. This plethora of information sets new challenges for modern technology which now has to deal with the effective management of the mass of data in order to ensure that reliable conclusions are reached rapidly and efficiently. Traditional web search engines are limited to text and image indexing and consequently many multimedia documents are excluded from these classical retrieval systems. Today, there are several systems able to perform searches on multimedia content, however they only allow queries based on the multimedia filename or nearby text on the web page containing the file and metadata embedded in the file such as title and author. Although this might yield some useful results if the metadata provided by the distributor is extensive, producing this data is a tedious manual task. It was precisely to address this need that a content-based search index for multimedia on the web search engine was built **[1]**. SPEECHBOT can achieve satisfactory accuracy of its transcriptions, if the acoustic and language models are properly trained.

Today, many radio and TV stations provide headlines, lists of keywords or even short summaries concerning the content of several audio or video news or talk shows. It is however, the provision of detailed electronic file indexing that will make multimedia file searches simpler and certainly much faster than the current time consuming process.

Other important applications of this task are speaker diarization and speaker tracking. In addition to improving ASR systems, audio segmentation has many other interesting and useful practical applications. Subsequent content based audio classification and retrieval have a wide range of applications across the entertainment industry in uses including for example audio archive management, commercial music usage, surveillance, etc. The audio libraries on the World Wide Web will surely employ audio segmentation indexing and searching. This is also an integral component of content-based indexing, archiving, retrieval and on-demand delivery of audiovisual content.

Great interest has been expressed by researchers in supplementing traditional minutes with audio summaries (M4 Project, 2002). Navigation through the meeting recordings or broadcast news archives with the aid of individual speaker speech segmentation would certainly improve and broaden the spectrum of information an interested user can have immediate access to. Particularly using these segmentation queues, one could directly access a particular segment of the speech made by a particular speaker and in situations like meeting recordings, information about presentations made by participants can be automatically and efficiently extracted.

Aside from this application reliably segmenting audio could be combined with a video shot detection system: shots that should have a significant audio novelty as video difference are more likely to be meaningful transitions. Another application might even be to play back an audio part synchronized with unpredictably timed events (such as progress through a video game).

Extracting information and performing segmentation on these speech files is challenging since such files consist of a wide variety of speaking conditions including clean speech, narrow-band speech, speech corrupted by music or background noises and music segments.

## 1.2 Summary of the problems examined

In this thesis we examine the event detection problem (detection of a speaker turn) in audio streams along with audio signal processing methods that extract specific audio characteristics.

Specifically, we study and implement the Bayesian Information Criterion (BIC). We concentrate on applying the widely known BIC method using several audio features as well as combinations of them in an effort to improve system performance. Furthermore, we propose an extension of BIC method to a Gaussian mixture distribution instead of the commonly used single Gaussian distributions. Finally, we investigate the use of transformation adaptation techniques such as Maximum Likelihood Linear Regression (MLLR) to transform the covariance matrices and expand the distance between the segments under examination.

## 1.3 Organization

- Chapter 2 discusses the event detection problem, the work done to date on several front-ends and segmentation methods and focuses on the advantages and disadvantages of the most commonly used segmentation methods and feature extraction employments.. We briefly summarize other state-of-the-art audio segmentation methods and proceed to analyzing the BIC criterion and its strong points in greater depth.
- In chapter 3, we present the front-end algorithms we consider produce appropriate features that could improve speaker and speaking condition modeling on a broadcast audio stream in conjunction with the BIC.
- Chapter 4 examines the experimental results and makes observations on every module, while comparing the results of each feature selection individually.
- In chapter 5, we fuse features and propose extending the BIC criterion. We also implement a new module combining various features.

- In chapter 6 we develop and utilize a Gaussian mixture and present the results of a new segmentation system.
- In chapter 7, we utilize the Maximum Likelihood Linear Regression for segmentation purposes. Difficulties and implementation issues are discussed and experimental results are shown.
- In chapter 8 we present our overall conclusions.

# Chapter 2

## Audio Segmentation

### 2.1 Problem Formulation

Research into audio segmentation is an ongoing process that focuses on partitioning an audio stream in terms of homogeneous regions. Homogeneity in our project is defined as audio segments that comprise the same speech, music, environment-background sound source or even silent segments with no sound. In short,, we are interested in detecting sections of the audio stream where two or more speakers are talking simultaneously and finally any music-to-speech, speech-to-silence and vice versa transitions. All the above constitute the problem of event detection (starting and endpoint) and consequently segmentation of the input audio signal.

Investigating and tracing these events on our input broadcast news signal would ideally result in classes in which there is:

1. No speech (silence, music or noise)
2. Only one speaker
3. Simultaneous talk of two or more speakers
4. Speech and music (e.g. commercials)

thus enabling us to define the problem on which this thesis focuses, as well as the error rates which are evidenced.

From point number 1, false alarms (insertions) and missed detections (deletions) can appear due to

- a relevantly small silence segment during one speaker utterance
- a variety of possibilities for the sound of the environment to change suddenly as a result of a live connection outside the studio
- periodic noise that can lead to us losing a speaker turn

From point number 2,

False alarms can occur, due to the change in the attitude of the speaker. It is possible that he may become angry and be considered a speaker turn, as well as misses due to a rapid change of speakers.

From point number 3,

Once again both errors can occur, since

a) a difference of opinion could cause Miss detection for the speaker who continues to talk immediately after the argument.

b) as speakers speak simultaneously with slightly more tension, one of the speakers could incorrectly be considered a speaker turn by the system.

Thus, we attempted to reduce the false alarm rate (false positives) and the Miss Detection Rate (false negatives) and ensure the accuracy of our detections.

Many approaches to this problem have been proposed in the international literature, some presented below. Most of them reformulate this task step-by-step. Essentially, initial segmentation is achieved by separating homogeneous speech and non-speech regions, followed by yet further segmentation into speaker turns or into environment alteration etc., according to the application. Hence there are many possible ways the given audio stream can be segmented.

The solution to this problem relies on extracting appropriate acoustic features-characteristics and arriving at the most suitable method to exploit those features which model the event detections we are targetting.

## 2.2 Previous Work

Content based audio retrieval has become vital in most audio based applications and consequently audio processing has gained a lot of interest and focus recently. In Section 2.2.1 we summarize the commonly used statistical and signal processing tools used in state-of-the-art audio processing techniques, for segmentation tasks. In Section 2.2.2 we present approaches that have been developed to date to address various problems and limitations of audio segmentation.

### 2.2.1 Acoustic Features

Human understanding of sound and the voice production system have been studied in order to evaluate characteristics that could even identify specific individuals who are entered onto a data base. This preprocessing module (feature extraction) is also referred to as “front-end” in the literature. A variety of methods are used to control the spectrum, amplitude, signal frequency content in an effort to extract significant data from audio files in order to further categorize the problem.

The most commonly used acoustic vectors are the Mel Frequency Cepstral Coefficients (MFCC), which are usually used along with their first or even their second derivatives. The characteristics of Linear Prediction Analysis (LPC) and Perceptual Linear Prediction (PLPC) based on the spectral information derived from a short time windowed segment of speech are also widely used, with the detail of the power spectrum representation being their main difference. MFCC features are derived directly from the Fast Fourier Transform (FFT) power spectrum, whereas the LPCC and PLPC use an all-pole model to represent the smoothed spectrum. The MFCCs follow the mel-frequency scale, giving greater detail at low frequencies as the filterbank centers and bandwidths above 500Hz are distanced further and further. LPCCs have an adaptive detail as the model poles move to fit the spectral peaks wherever they occur. This detail is limited by the number of poles available. PLPCs use both filterbank and all-pole model spectral representation. Firstly, they follow the bark-spaced trapezoidal filterbank and then it is fitted with an all-pole model. Finally, the spectral representation is transformed to cepstral coefficients. There have been many variations of MFCCs, like Teager Energy Cepstral Coefficients (TECC) based on teager energy and they are more robust in noise **[2]**.

Several one-dimensional features are widely used as mean square amplitude or root mean square amplitude (RMS), maximum amplitude (envelope), short time energy (STE), zero-crossing rate (ZCR) and many variations of them. Alternatives have been developed that are based on Energy separation algorithm (ESA) **[3]**, which use this algorithm along with signal's teager energy. Maximum average Teager Energy (MTE), Mean Instantaneous Amplitude (MIA) and Mean Instantaneous Frequency (MIF) are some of them.

Finally prosodic features are recently widely used as they may offer further information of the audio signal, besides usual cepstral coefficients. Speaker information may be found in both static and dynamic forms and may originate from anatomical, physiological, or behavioral differences among individuals. Such features are the fundamental frequency (f0) and energy.

### 2.2.2 Segmentation Methods

Segmentation is a key process since the subsequent (usual) clustering process depends to a great degree on the quality and homogeneity of the segments obtained. Due to the importance of audio parsing algorithms for speech processing, a number of approaches have been proposed over recent years. In this section, we will review a number of typical approaches. These algorithms are ranked into the following categories:

1) Metric based
2) Gaussian mixture model (GMM) based
3) Recognition based
4) Model selection based
5) Hybrid approaches

## *1) Metric based segmentation*

Metric based algorithms rely on usual pattern recognition problems where classes are distinguished through distance estimation methods, i.e. how far class one is from the mean value of all observations compared to the distance of class two (metrics are the mean and variation of observations). So a straightforward method of audio segmentation is to detect acoustic change points based on spectral changes. The underlying assumption is that the data of different acoustic types possess different spectral shapes, and these differences can be sufficiently measured by the distances between the acoustic feature vectors. In practice, the spectral changes are identified at the maxima of the dissimilarity in terms of some metric between neighboring windows that shift along the audio stream. Here, a window is typically two seconds, which should be longer than a speech frame. The choice of an appropriate distance measure is essential to segmentation performance for this class of algorithms. Previous studies have introduced the use of the Generalized Likelihood Ratio **[4]** and the symmetric Kullback-Leibler distance (KL2) **[5]** as the distance metric. If the windowed observations are modeled by the multivariate Gaussian distributions N ($\mu_1$, Σ1) and N ($\mu_2$, Σ2), then the KL2 distance **[6]** between these two neighboring windows is defined by

$$\mathrm{KL2}_{1,\,2} = (\mu_1 - \mu_2)' \, (\Sigma_1^{-1} + \Sigma_2^{-1}) \, (\mu_1 - \mu_2) + \mathrm{tr} \, (\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I) \quad (1)$$

Such a distance measurement is continually calculated between neighboring windows along the audio stream and a distance curve is formed. To avoid fluctuations due to noise-corrupted speech, this curve is often smoothed using a low-pass filter. The local peaks over the curve can be treated as candidate segmentation points. However, it is often difficult to determine the final segmentation points from these candidates since this requires suitable thresholding, which is often tuned from training data, but cannot guarantee stability and robustness for all test data. A variation of K-L distance is the divergence shape distance (DSD), (**[7].[8]).**

Another usual strategy is the Log Likelihood Ratio criterion (LLR) which is used in **[9]** and a variation of it (LLRC) referenced in **[10]** that leads to an improved performance.

Furthermore, a criterion based on Vector Quantization technique is described in **[8]**. The VQ approach is based on the generalized distance between two feature vectors, typified as $S^A$ and $S^B$. The VQ distortion measure between $S^B$ and codebook $C^A$ produced by grouping the features of $S^A$ is defined as:

$$VQD(C^A, S^B) = \frac{1}{T}\sum_{t=1}^{T} arg \min_{1\leq k\leq K}\{d(C_k^A, S_t^B)\}$$

Where $C_k^A$ represents the k-nth code-vector in $C^A$, $S_t^B$ is the t-nth feature vector of $S^B$ and d is the Euclidean distance.

To conclude with metric based segmentation, a new distance measure criterion is the Weighted squared Euclidean Distance (WED) proposed in **[11]**. This criterion is based on the Euclidean distance between feature vectors in frame comparison, while it uses weights that depend on the variation of characteristics.

### *2) Gaussian Mixture Model Based Segmentation*

Gaussian Mixture models have been thoroughly studied and consequently widely used since they can model any set of observations with unknown probability density function. In a simple classification problem of two classes the fastest way to decide between the two hypotheses is the fraction of their likelihoods given an experimentally estimated threshold. Suppose $H_0$ is the hypothesis that our observations belong to speech stream and H$_1$ to non-speech stream, according to above we get:

$$\frac{p(Y|H_0)}{p(Y|H_1)}\begin{cases}\geq \theta\ , accept\ H_0\\ < \theta\ , accept\ H_1\end{cases}$$

We define the model $\Lambda(X)$ to represent the above hypotheses as a function:

$\Lambda(X) = \log p\big(X|\lambda_{hyp}\big) - \log p\big(X|\lambda_{\overline{hyp}}\big),$

X is the observation vector and $\lambda_{hyp}$ is the class being tested.

This formula is preferred due to the relative simplicity of the calculations. Next we must decide the probability density function $p(X|\lambda)$ that best discriminates and determines each class. With the given GMM, the observation vector $\chi_t$ at time t is distributed as:

$p(\chi_t, \Lambda) = \sum_{m=1}^{M} w_m p_m(\chi_t\ \Lambda_m)$, where

$$p_m(\chi_t\ \Lambda_m) = N(\chi_t; \mu_m, \Sigma_m)$$

n is the feature vector dimension M is the number of mixtures $w_m$ is the mixture weight of the m$_{th}$ component of the mixture Gaussian with the constraint that $\sum_{m=1}^{M} w_m = 1$ and $\mu_m$ and $\Sigma_m$ are the mixture mean and covariance. The number m of components used to approximate the observation distribution depends on the sensitivity our system needs to be trained.

Following the introductory method to apply a GMM for audio segmentation **[12]**, several sets of Gaussian mixture model parameters are estimated for the K classes of different acoustic conditions from the training data. Expectation-Maximization **[13]** is the most widely used algorithm to train the GM models. Then the observation features of the processing audio stream are classified into one of the K classes which when “compared” to others, result in the maximum likelihood. The segmentation is decided at locations where acoustic condition changes.

Obviously this approach is less practical, because it requires pre-trained GMMs and a priori knowledge of acoustic conditions in order to define the classes. Consequently this method is usually used as a pre-process step of segmentation, to identify speech and non-speech turns.

### *3) Recognition-Based Segmentation*

These methods use the prior knowledge of existing classes of the audio streams and use classification techniques to produce the segments. Hain et al **[14]** proposed a multi-pass decoding process to perform the segmentation. The audio classification uses Gaussian mixture models (GMM) with 1024 mixture components and diagonal covariance matrices which are then decoded using a conventional Viterbi search over a network of the four trained states each of which model the a) wideband speech b)telephone speech c) music or non-speech  and d) speech and music, so the stream consists of these four classes now, with an inter-class transition penalty to prevent frequent transfer between states and thus to produce longer segments. To improve frame classification accuracy, the underlying acoustic models of these 4 states are dynamically adapted using Maximum Likelihood Linear Regression (MLLR). Next, pure non-speech segments are discarded, and others are decoded through another round of gender-dependent phone recognition. The phone recognizer contains 45 context independent phone models per gender plus a silence/noise model with a null language model. The output is a phone sequence with male, female or silence tags. The phone tags are ignored and the phone sequence with the same gender label is merged. A set of heuristic rules are further applied to smooth the gender boundaries. Finally, the change points between genders are marked and results to segments by gender transitions. It is obvious  that this method requires a relatively complicated flow process for segmentation. Furthermore, the general disadvantage is that it is unable to detect speaker transitions between two speakers of the same gender when there is no significant intervening silence. This will be questionable for speaker turn segment detection.

A recognition based segmentation method is the LIMSI system **[15]**. Again they use Viterbi decoding and GM modeling to segment the input audio stream in speech, speech with music or noise, music, noise or silence. Then in speech sections a metric-based segmentation is employed and each of the resulting segments is trained. Next an iterative GMM segmentation \ clustering procedure follows in order to produce homogeneous classes alternating Viterbi decoding – GMM re-estimation  and merging similar classes. Essentially the speech sections are separated through the maximization of:

$\sum_{i=1}^{N} \log \mathrm{f}(s_i|M_{c_i}) - \alpha N - \beta K$,

Where S= ($s_1$,…,$s_N$) is the speech sections split in N segments, $c_i$ is the label for each $s_i$ segment (among K classes), $f(s_i|M_{c_i})$ is the probability of $s_i$ given the model $M_{c_i}$ and α, β are penalties for the segments and classes respectively.

To determine the final detailed bounds of each segment a new Viterbi decoding is employed, using energy based restriction. Eventually appropriate GMMs are trained to categorize the final segments in Bandwidth and Gender Identification. A diagram of the above operations follows.

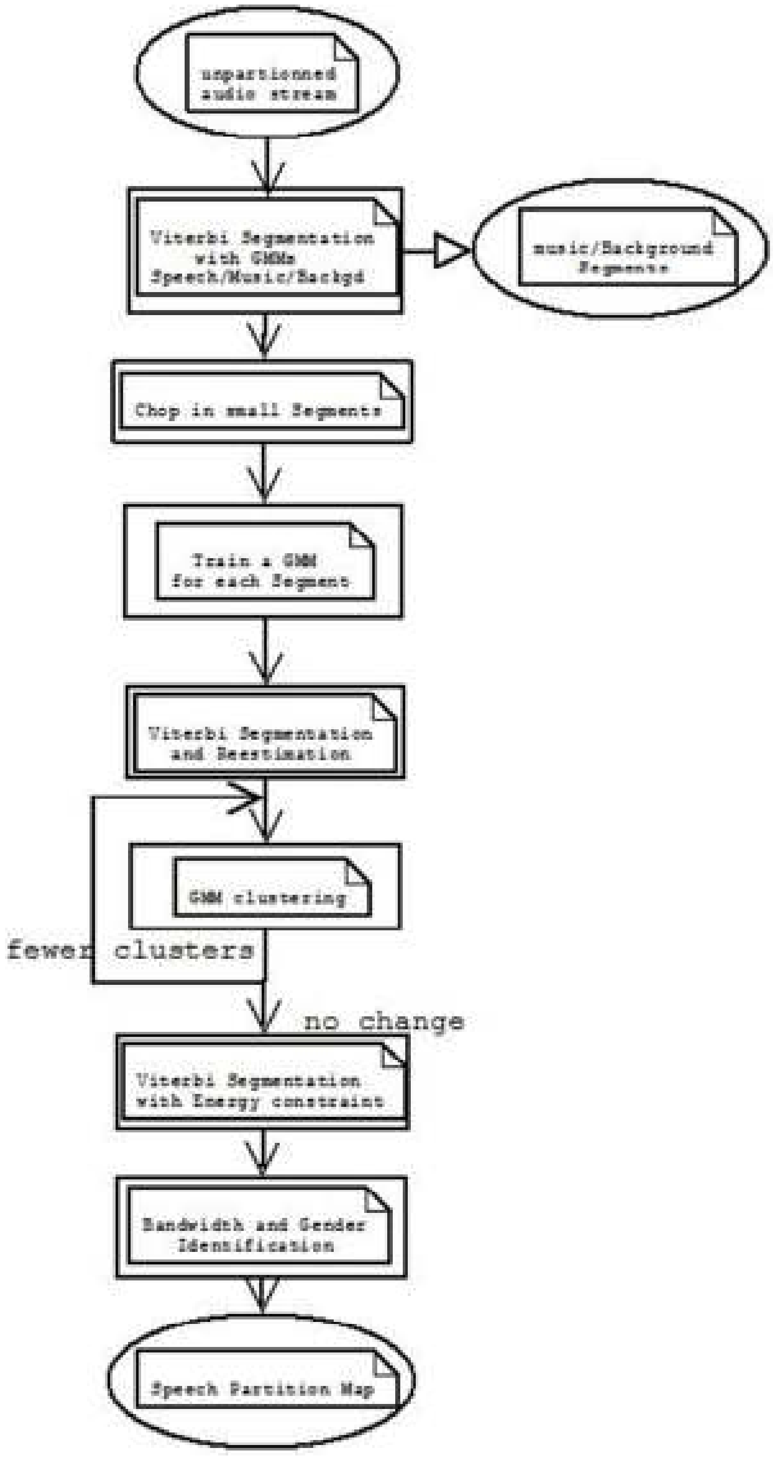

unpartionned audio stream
Viterbi Segmentation with GMMs Speech/Music/Backgd
music/Background Segments
Chop in small Segments
Train a GMM for each Segment
Viterbi Segmentation and Reestimation
GMM clustering
fewer clusters
no change
Viterbi Segmentation with Energy constraint
Bandwidth and Gender Identification
Speech Partition Map

***Image 2.1: The LIMSI system***

### *4) Model selection segmentation*

An alternative approach to audio segmentation methods is proposed by Chen and Gopalakrishnan **[16]**. In their study, the segmentation problem is reformulated as a model selection task between two nested competing models. This method employs the Bayesian Information Criterion (BIC) as the model selection criterion, illustrating several desirable properties such as robustness, threshold independence and optimality. BIC is a penalized maximum likelihood model selection criterion that has been widely used in statistical data processing. With such a scheme, the segmentation decision is derived by comparing BIC values. Other advantages of this scheme include that no prior knowledge concerning acoustic conditions is required and no prior model training is needed. Numerous variations of this criterion exist in literature, in the effort to further improve its good performance as described in **[17-28].** This approach is further described in **3.1** as we implement our segmentation system based on it.

### *5) Hybrid methods*

In the effort to improve segmentation results, there have been implemented systems using a combination of the previously mentioned methods along with other ideas, i.e. neural networks.

In **[29]** a system based on LIMSI is presented where instead of GMM segmentation \ Clustering procedure stage, a Bayesian clustering is applied. During this step, classes are merged depending on the BIC values and stops when ΔBIC is greater than or equal to zero. Next an optional clustering stage is applied to categorize in further detail the BIC resulting classes and a final process to discard likely existing silence segments.

Moreover, a combination of neural networks with Hidden Markov Models (HMMs) system is proposed in **[30]**. Here Multi-Layer Perceptron is used to estimate the posterior probabilities of speech phones on a fixed length frame of feature vector. Dynamism and Entropy characteristics are estimated using the above probabilities and they are used as input in HMMs, which classify the audio stream in music and speech segments. A fully associated two state HMM is employed image 2.2 (music and speech) with each state derived by successive sub-states of similar context modeled by Gaussian mixture models.

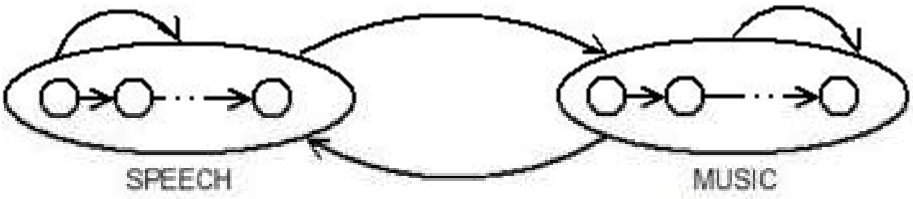


*Image 2.2: Two state HMM*

Finally, an alternative approach that combines several methods is proposed in **[31]**. Firstly, non-speech and speech classes ensue, using widely pattern recognition methods as K-nearest neighbor algorithm (KNN) and linear spectral pairs vector quantization (LSP-VQ). Secondly, the non-speech classes are further categorized in silence, noise and music sub-classes using energy-based methods after a new set of specialized features are extracted. The speech classes are typically processed by GMMs. This system may be used on real-time applications as it performs well and due to its good consumption factor.

# Chapter 3

## Bayesian Information Criterion – Front-End Research

### 3.1 BIC Analysis

In this section we describe a maximum likelihood approach for acoustic change detection based on the BIC, a criterion penalized by the model complexity (the input parameter dimension) which has the advantage to model-decoding-based segmentation that does not require *a priori* knowledge of the input audio stream. This method was first proposed by **[16]** and it is assumed that the acoustic feature vectors in each of the two audio segments are drawn from a Gaussian distribution and a change detection results from the dissimilarity of the Gaussians. Indeed, we determine if our dataset is modeled by two Gaussians or just one Gaussian.

Given the feature vectors (MFCCs as a baseline implementation) X = {$x_i$ :I =1, … , N} we propose two models for our data. The null hypothesis $H_0$, which states that all feature vectors are independent and identically distributed (i.i.d.) samples drawn from the same Gaussian $N(X;\mu_0,\Sigma_0)$ , while in the alternative hypothesis $H_1$ the first $x_i$ vectors $x_1 \ldots x_i \sim N(\mu_1,\Sigma_1)$ and $x_{i+1} \ldots x_N \sim N(\mu_2,\Sigma_2)$. We estimate the parameters of the Gaussians from the data themselves and then we use the maximum likelihood ratio statistic which is:

$$R(i) = N \log|\Sigma| - n_1 \log|\Sigma_1| - (N - n_1) \log|\Sigma_2|$$

Where Σ, $\Sigma_1$ and $\Sigma_2$ are the covariance matrices from all the data,

from {$x_1$ , … , $x_i$ } and from {$x_{i+1}$ , … , $x_N$} respectively. The difference between the above models can be expressed as:

$$\text{BIC (i)} = \text{R (i)} - \lambda\text{P}, \qquad (2)$$

Where R(i) is defined in (1) ,the penalty P is:

$$\text{P} = {}^{1}\!/_{2}\left(d + {}^{1}\!/_{2}\, d(d+1)\right) \log N$$

And the penalty weight λ = 1; in detail λ depends on data so we use it empirically to express our threshold for (2), d is the dimension of our Gaussians.

To conclude with BIC if (2) is positive then the model of two Gaussians is favored and thus we detect a change in environment, channel and speaker identity.

$$\{MaxBIC\ (i)\} > 0$$

Implementation pseudo code:

```
   While Not End of File {
   If (flag==0)  {          %Investigating event detection
     Read audio file from-to;
            }
   If (flag==1) {           %No  event  detection,  increase  the
window
     Read audio file from-to+BiggerWindow;
            }
   Extract features from the input signal;
BIC = ComputeBIC(ceps,step,pad); %iterative  estimation  on  each
                           frame  of  the  %coefficients.1st  with  the
                           rest, next, the two
                           %first with the rest until the end.
[maxBIC,maxIndex] = max(BIC);
   If (maxBIC > threshold) {     #Event Detection
     from = from + maxIndex*frame_length;
flag=0;
else {
     BiggerWindow = BiggerWindow+increase;
     Flag=1;
     }
End
```

Our implementation was initially based on a minimum miss of real change detection rate and empirically turned out to use the 100milliseconds frame with 50milliseconds overlap from our baseline feature vector.

We had to choose a window size that would limit events but also be wide enough to give us the opportunity to process the audio stream relatively fast. If change is distanced too far from the previous one, it takes too long to process our input audio stream since we need to evaluate the determinants of *two full covariance matrices for every possible break point in a window*. Focusing therefore on the problem of multiple change detection we needed an increase in step window size when there is no change detection. Taking into consideration the above, we selected a two second increase step.

Subsequently to all the above being tested, we reached an ideal threshold equal to 202 for the specific setup. Eventually, when a change point is detected (BICvalue>202) the ten-second-window is applied starting from the end of the previous segment. While there is no change we increase the window size by two seconds until BIC exceeds our threshold and a new segment is found.

Having reached a point at which we are ready to evaluate our system we must mention that a change point that is detected beyond 600 milliseconds is clustered in miss detection rate. 500 ms justified by precision of the implementation and the extra 100ms by the hand-segmented data, because we emphasize the speaker turn events rather than complete silence segments. The results of our experiments are presented in **section 4.**

## 3.2 Feature Selection-Extraction

In the effort to produce characteristics, that would offer solutions to problems encountered in ongoing research like automatic speech recognition, automatic speaker segmentation, speaker identification-verification, language identification telecommunications etc., the procedure of audio signal processing has been studied over many years. Focusing on each of the aforementioned problems, we identified several common features and applied them to our specific problem.

Generally, the speech signal is not a stationary stochastic signal, but can be regarded as such if we split the signal into small enough pieces. Windowing is precisely that, the splitting of the signal into smaller (usually 10-50 ms) parts, with an overlap between the windows, ensuring that there is no loss of information at the borders. Much help over speech signal processing may be found in **[32].** We extract features from each window that are able to model our signal and be used in our experiment. Research of the existing international bibliography reflects that that short-term spectral information offers very good results for our purpose, thus at least for our baseline measurements we should certainly begin with such features. We therefore use Mel frequency cepstrum coefficients, the most widely used speech features, as our baseline and then search for alternative coefficients which we then compare to our baseline and attempt to identify novel combinations to produce improved results. Next in this chapter we present the one- and multi-dimensional alternative characteristics, their applicability, the way they model the audio signal and which methods are employed to extract each of them from the audio signal.

### 3.2.1 Mel Frequency Cepstral Coefficients (MFCCs)

Mel frequency cepstrum coefficients, whose function and performance have proven good results on every level of speech processing like audio segmentation and speech recognition, are derived by the log-energies in frequency bands distributed over a Mel scale. This distribution of frequency bands is preferred because the human perception of tone frequency doesn’t follow linear scale.

The formula to convert f hertz into m Mel is:

$$m = 2595\log_{10}\left(1 + \frac{f}{700}\right) .$$

MFCCs are commonly derived as follows:

1. Take the Fourier transform of a windowed frame of a signal.
2. Map the powers of the spectrum obtained above onto the Mel scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the Mel frequencies.
4. Take the discrete cosine transform of the list of Mel log powers and keep the first N coefficients.
5. The MFCCs are the N coefficients for each frame.

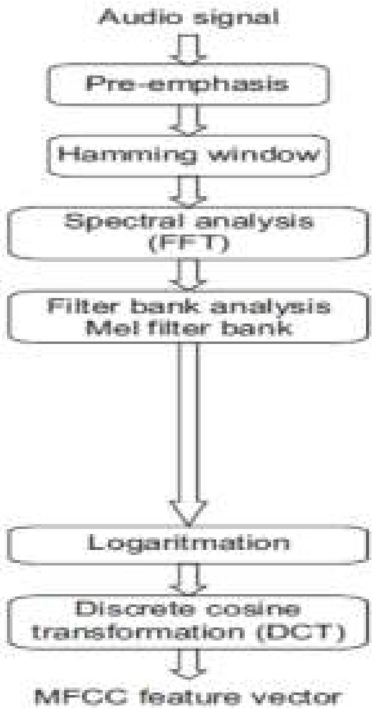


***Image 3.1: Mfcc's extraction process***

Based on our experiments with the MATLAB VOICEBOX we came up with 10 MFC coefficients. Forty filters were employed for feature extraction and triangles centered on each of the equally distributed frequencies over the mel scale.

Experimentally, based on a fixed threshold initially, we observed that a greater number of MFCCs led us to over-segmented audio streams. Moreover there is a drawback on the number of coefficients that can be used, meaning that many more observations must be used in order to train our Gaussian, while the number of coefficients increases. Over-segmentation was further encouraged with small frames and overlaps, so empirically we tested 100ms with 50 ms overlaps (from here on referred to as frames with overlap as 100_50ms), 50 ms and 40 ms frames. Naturally, wider frame lengths extended the Miss Detection Rate (Number of real changes that were not detected).

We will see in further detail, in chapter 4.4 the comparisons of frame lengths over several thresholds and with varying BIC window-size tests. Below the behavior of BIC through Mfcc's is represented.

***Voice signal***

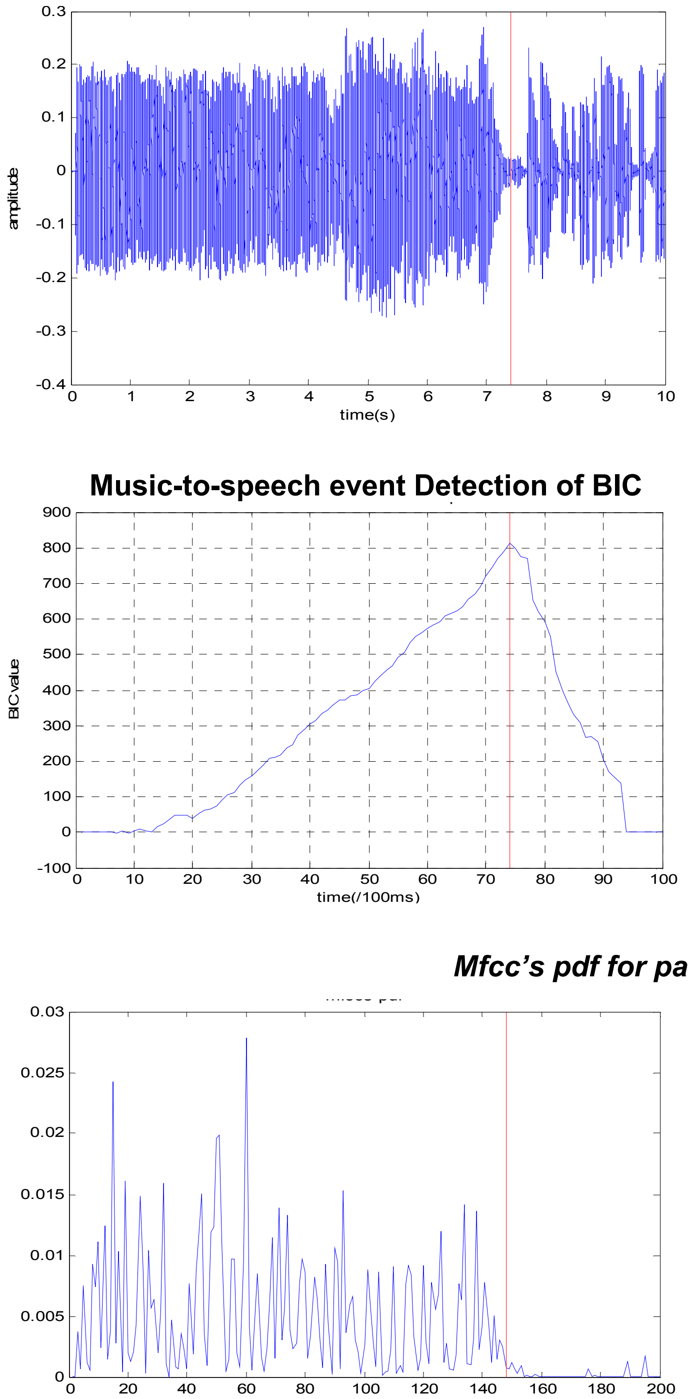


**Figure 3.1** On the first image the audio signal of a transition from music to speech appears (at 7.4 sec). Then the BIC values are presented every 100ms for these 10 seconds (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix) and finally the modeling of the signal by the MFCCs.

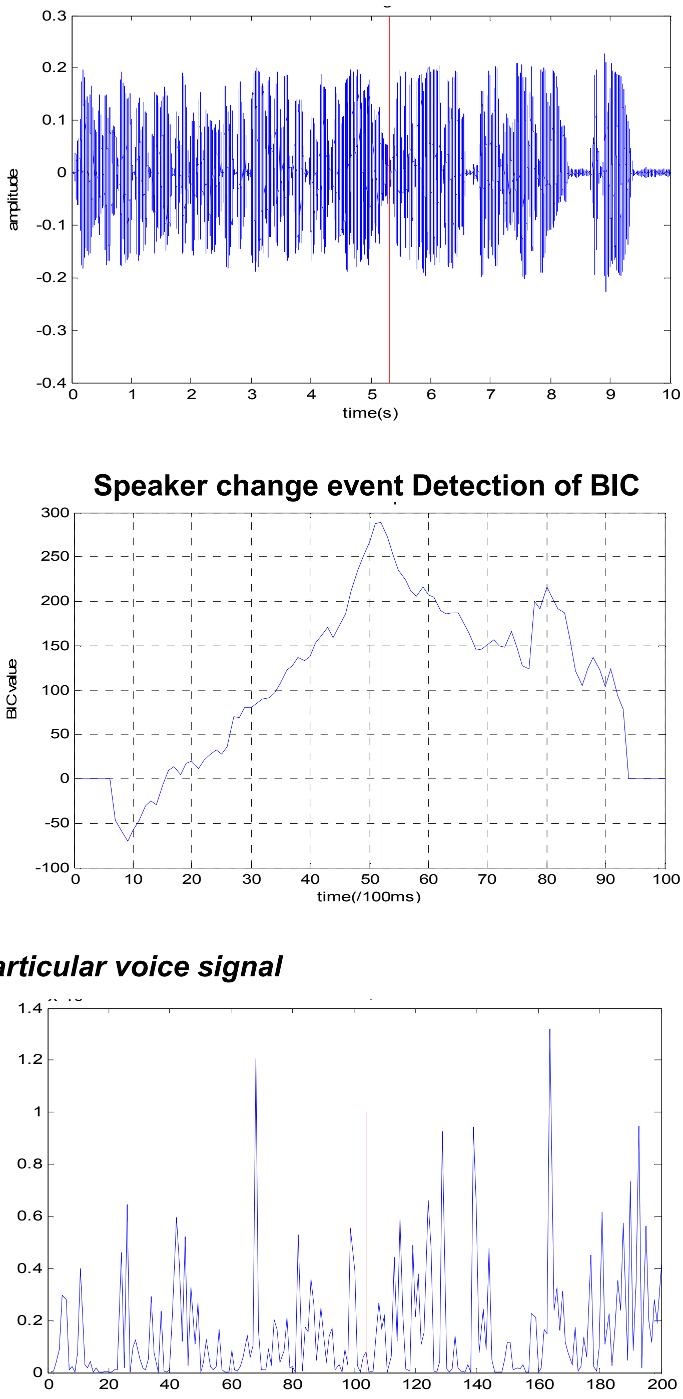


**Figure 3.2** On the first image the audio signal of a speaker change appears (at 5.3 sec). Then the BIC values every 100ms are presented for these 10 seconds (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix) and finally the modeling of the signal by the MFCCs

### 3.2.2 Perceptual Linear Predictive parameters

The Perceptual Linear Predictive (PLP) parameters are also widely utilized in speech recognition. The PLP method **[33]** is mainly based on the findings from the research of psychoacoustics. Thus, PLPs result from standard all-pole modeling, or linear predictive analysis of a specially modified, short-term speech spectrum. In PLP the speech spectrum is modified by a set of transformations that are based on models of the human auditory system. These coefficients are often used because they correctly approximate the high-energy regions of the speech spectrum while simultaneously smoothing out fine harmonic structure, often characteristic of the individual but not of the underlying linguistic unit. The spectral resolution of human hearing is roughly linear up to 800 or 1000 Hz, but it decreases with increasing frequency above this linear range. PLP incorporates critical-band[1] spectral-resolution into its spectrum estimate by remapping the frequency axis to the Bark scale and integrating the energy in the critical bands to produce a critical-band spectrum approximation. At conversational speech levels, human hearing is more sensitive to the middle frequency range of the audible spectrum. PLP incorporates the effect of this phenomenon by multiplying the critical-band spectrum by an equal loudness curve that suppresses both the low and high frequency regions relative to midrange from 400 to 1200 Hz.

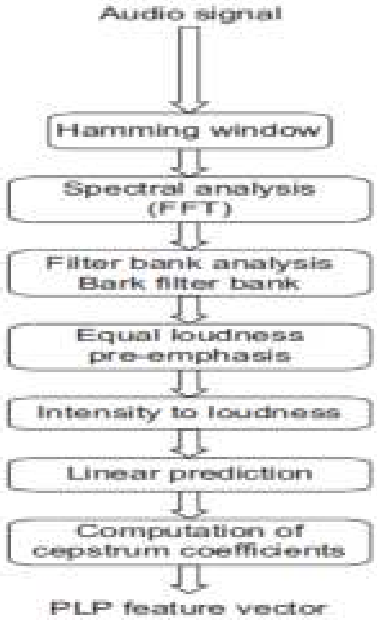
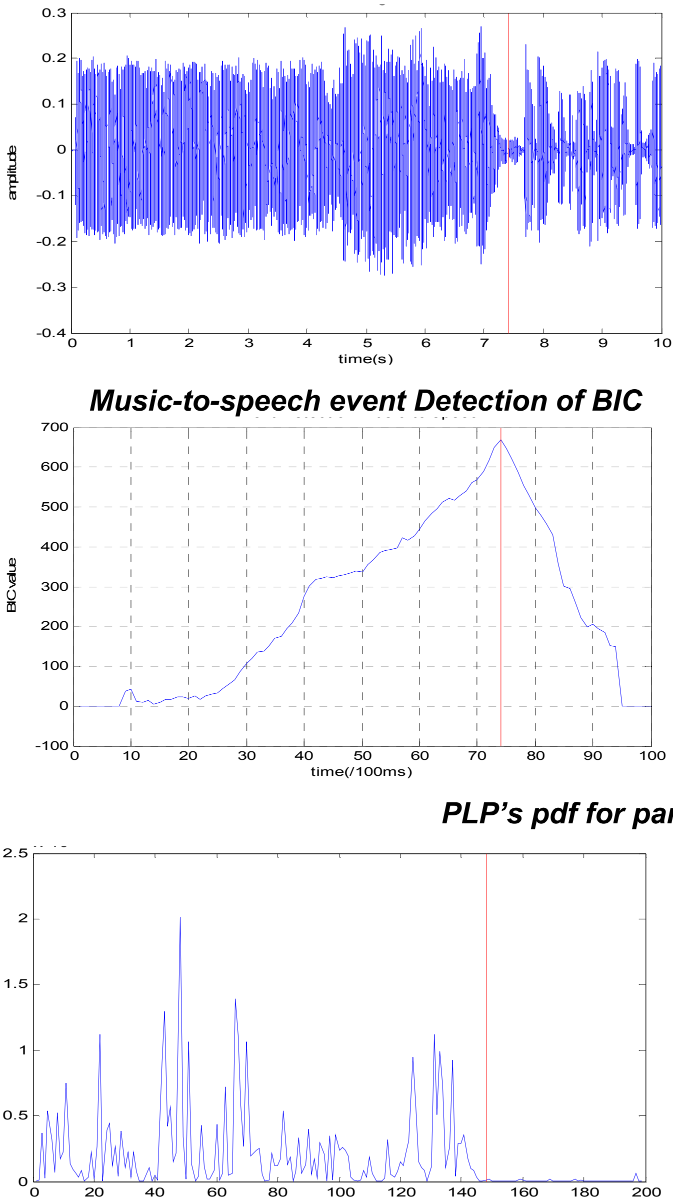


***Image 3.2: PLP's extraction process***

Starting with 10 MFCCs, we extracted similar dimension PLP parameters over the same frame length range and BIC thresholds in order to get a complete comparison of the features. Characteristic figures follow to show their performance and their fit in a Gaussian distribution.

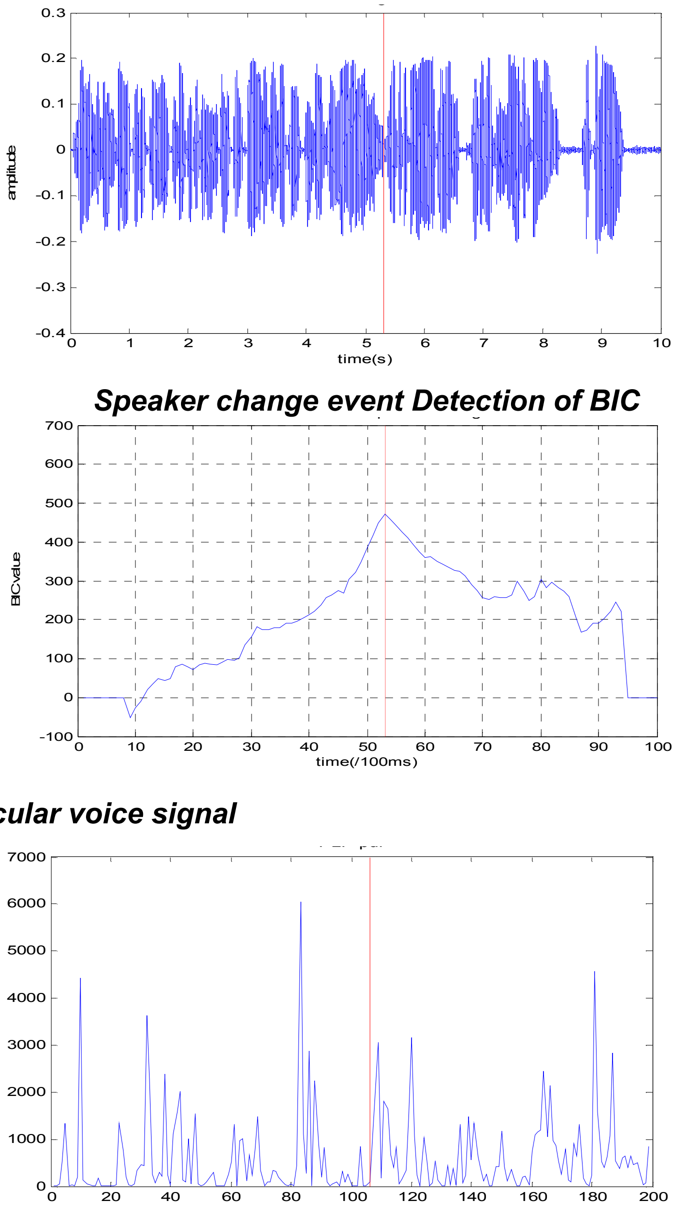
***Voice signal***

***PLP's pdf for particular voice signal***



***Music-to-speech event Detection of BIC***

**Figure 3.3** On the first image the audio signal of a transition from music to speech appears (at 7.4 sec). Then the BIC values are presented every 100ms for these 10 seconds (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix) and finally the modeling of the signal by the PLPs.



***Speaker change event Detection of BIC***

**Figure 3.4** On the first image the audio signal of a speaker change appears (at 5.3 sec). Then the BIC values are presented every 100ms for these 10 seconds (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix) and finally the modeling of the signal by the PLPs.

### 3.2.3 RASTA-Perceptual Linear Predictive parameters (RASTA-PLPs)

Frequently, the speech parameter estimators are greatly influenced by the frequency response of the communication channel. A technique that is more robust to such steady-state (something that we would like to take advantage of) spectral factors in speech is the RASTA technique **[33]**, which applies a bandpass filter to each spectral component in the critical-band spectrum estimate. This filtering emphasizes frame to frame spectral changes that occur between the rates of 1 to 10 Hz. Before applying the bandpass filter, log-RASTA takes the natural logarithm of each spectral component and this logarithm then converts multiplicative distortions in the frequency domain into an additive distortion, which can be filtered. Conversion to the log-spectrum domain is a common approach used in signal deconvolution problems. The rate of change of nonlinguistic components of speech and background noise environments often lies outside the typical rate of change of vocal tract shapes in conversational speech. Also, informal studies showed that the human hearing system is relatively insensitive to gradually varying stimuli. The basic idea of RASTA filtering is to exploit these phenomena by suppressing constant and gradually varying elements in each spectral component of the short term auditory-like spectrum prior to computation of the linear prediction coefficients. Thus RASTA highpass filtering removes gradually varying components in each element of the filter-bank output, such as those introduced by communication channels and RASTA lowpass filtering removes rapidly changing components typical of changes that are not phonetically important.

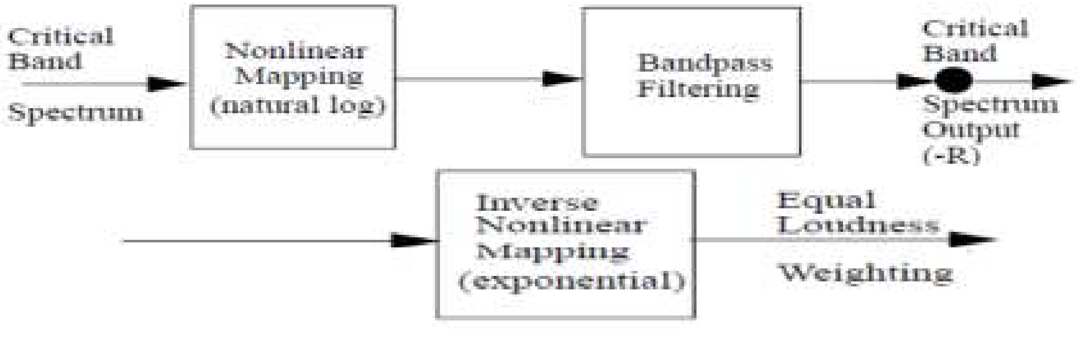
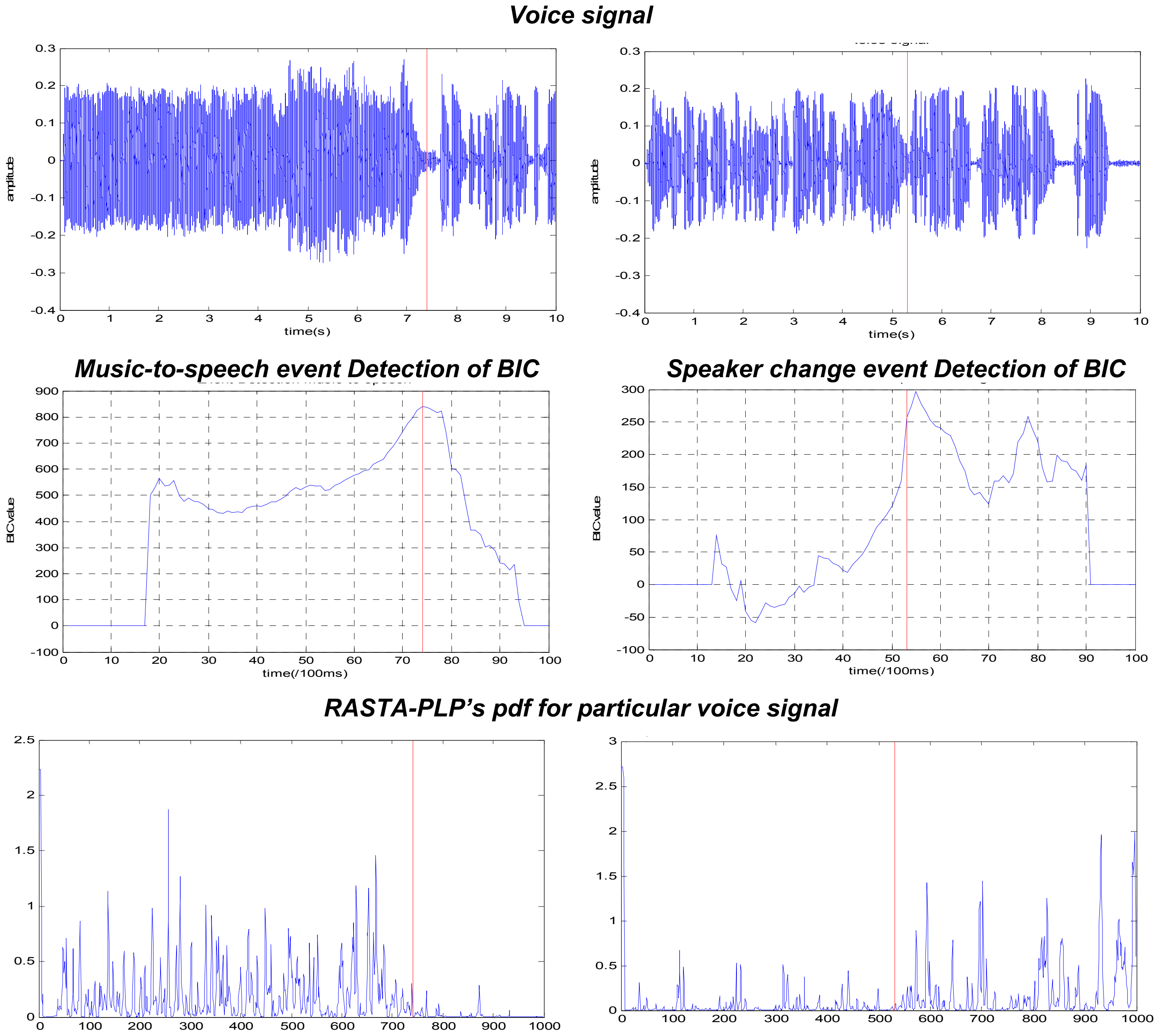

Critical Band Spectrum
Nonlinear Mapping (natural log)
Bandpass Filtering
Critical Band Spectrum Output (-R)
Inverse Nonlinear Mapping (exponential)
Equal Loudness Weighting

***Image 3.3: RASTA_PLP's extraction process***

Following the same rationale to compare our characteristics, 10 R-PLPs were extracted over the same frame lengths. Observing their behavior and the increased Miss detection and False alarm rate we employed two techniques to process the features before estimating the BIC values. "Averaging" and Cepstral Mean Normalization (CMN) that will be discussed further on.



***Voice signal***

***Music-to-speech event Detection of BIC***

***Speaker change event Detection of BIC***

***RASTA-PLP's pdf for particular voice signal***

**Figure 3.5** On the first image the audio signal of a transition from music to speech appears (at 7.4 sec). Then the BIC values every 100ms for these 10 seconds are presented (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix) and finally the modeling of the signal by the Rasta_plp's.

**Figure 3.6** On the first image the audio signal of a speaker change appears (at 5.3 sec). Then the BIC values every 100ms for these 10 seconds are presented (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix). This estimation fell out by 200ms which is not considered as False alarm. Finally the modeling of the signal by the Rasta_plp's is depicted.

### 3.2.4 Shifted-Delta Cepstral Coefficients (SDCC)

Previous studies **[34]** have shown that improved speaker and language identification performance can be obtained by using the shifted delta cepstral (SDC) feature vectors, which are created by stacking delta cepstral computed across multiple speech frames. The SDC features are specified by a set of 4 parameters, *N, d, P, k,* where ***N*** is the number of cepstral coefficients computed at each frame, ***d*** represents the time advance and delay for the delta computation, ***k*** is the number of blocks whose delta coefficients are concatenated to form the final feature vector and ***P*** is the time shift between consecutive blocks. Accordingly, *kN* parameters are used for each SDC feature vector, as compared with *2N* for conventional cepstra and delta-cepstra feature vectors. For example, for the case shown in figure below the final vector at frame time *t* is given by the concatenation of all the $\Delta c(t + iP)$ , where

$$\Delta c(t) = c(t + iP + d) — c(t + iP — d)$$

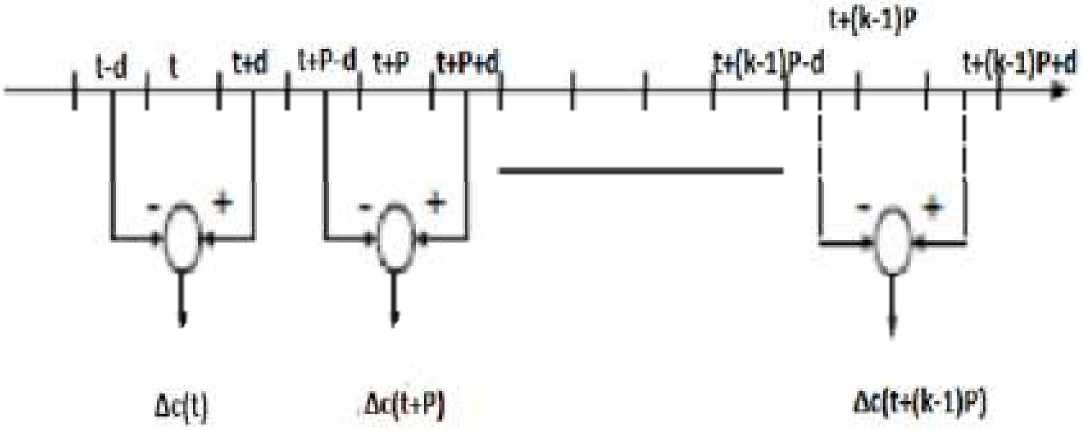


***Image 3.4: Shifted Delta coefficient's extraction process***

We implemented the SDCCs with parameters N=10, d=1, P=3, k=2. These characteristics ensured extremely high sensitivity to our system, thus leading to hyper segmentation of the audio signal.

***Voice signal***

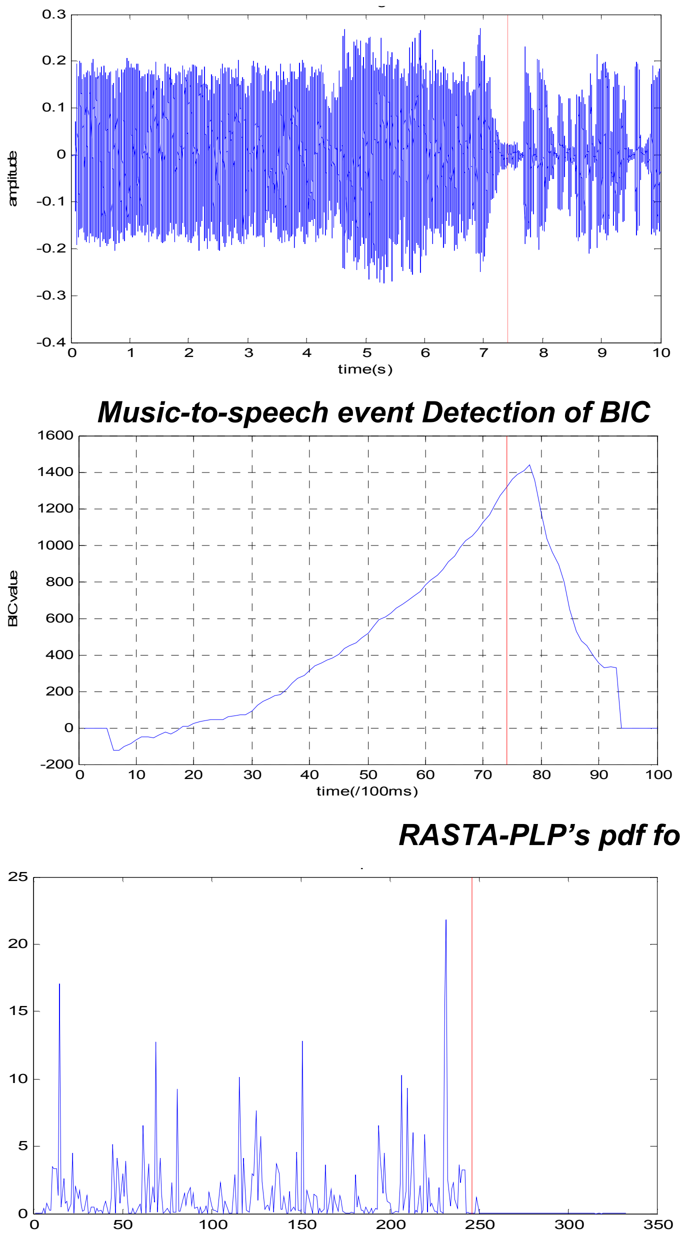


**Figure 3.7** On the first image the audio signal of a transition from music to speech appears (at 7.4 sec). Then the BIC values every 100ms for these 10 seconds are presented (excluding 1 sec at start and 1 sec at the end needed to estimate the determinant of the covariance matrix).This estimation fell out by 400ms and is not considered as a false alarm. Finally the modeling of the signal by the SDCCs is depicted.

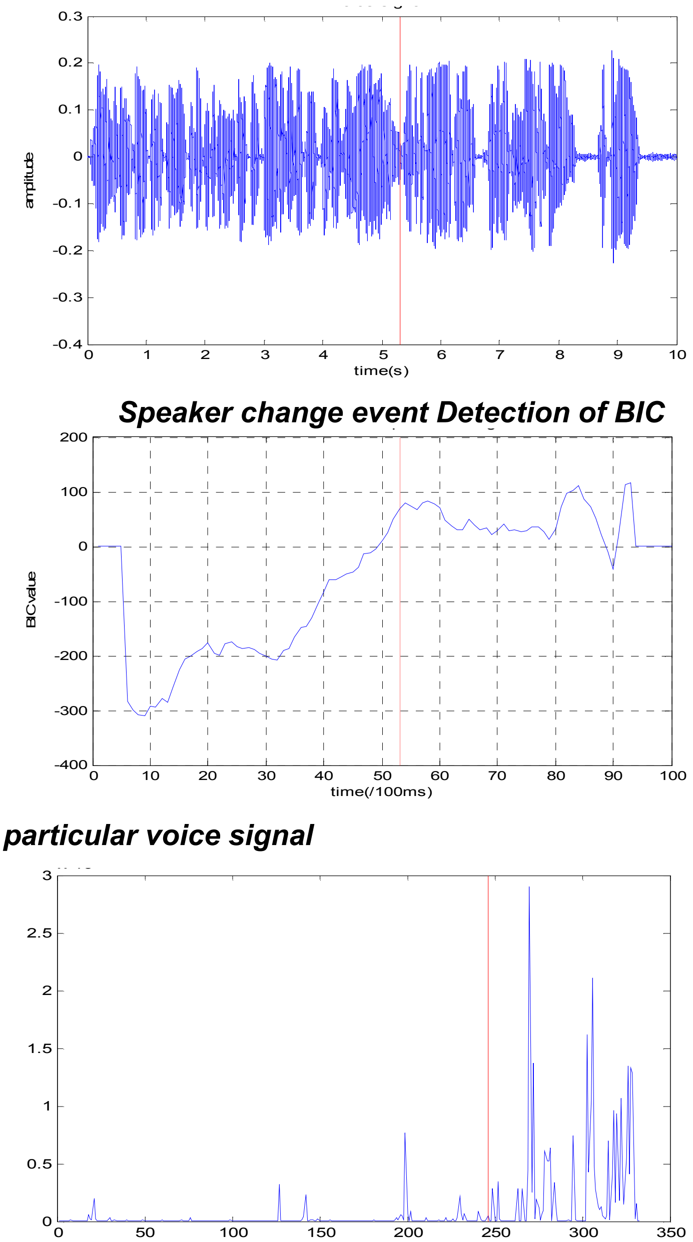


**Figure 3.8** On the first image the audio signal of a speaker change appears (at 5.3 sec). Then the BIC values every 100ms for these 10 seconds are presented (excluding 1sec at start and 1sec at the end needed to estimate the determinant of the covariance matrix). This estimation lost the event and is measured as miss detection, moreover a false alarm was created later. Finally the modeling of the signal by the SDCCs is depicted.

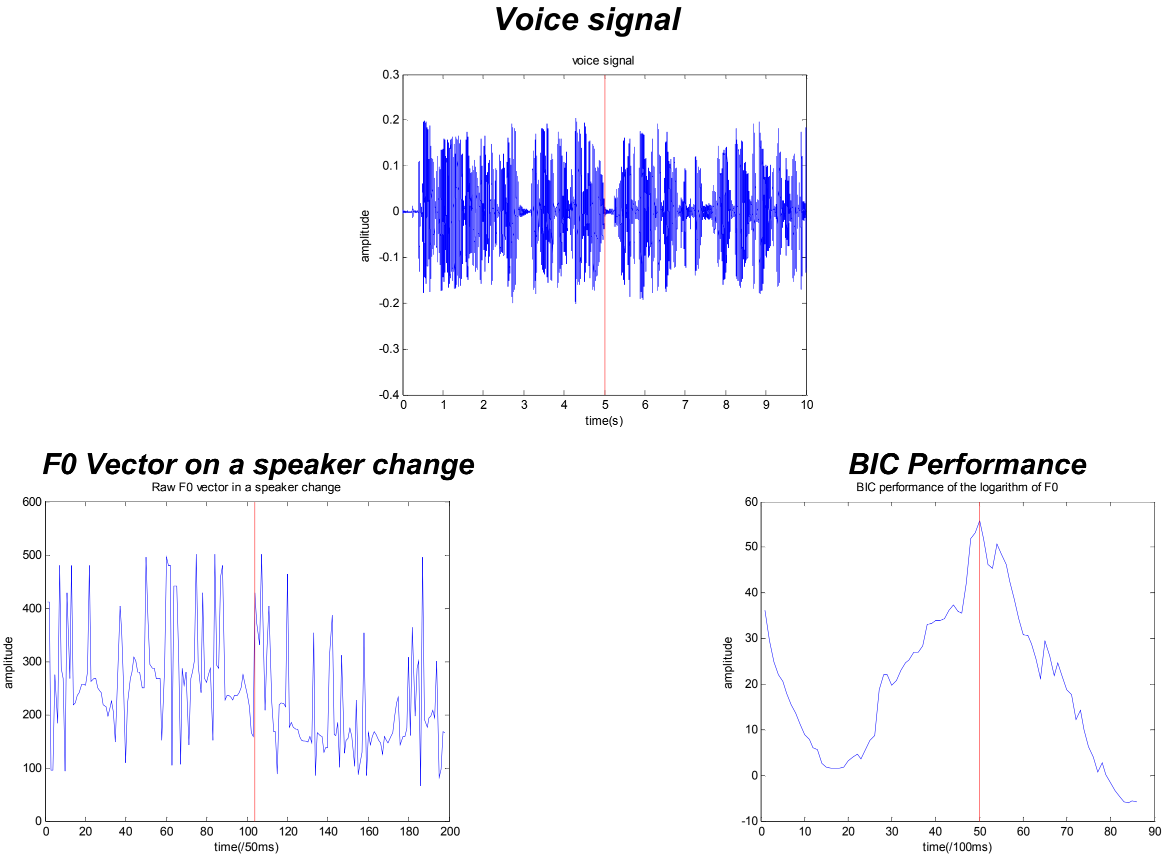
### 3.2.5 Prosodic Features

Current study **[35]** shows that long-term information can convey supra-segmental information, such as prosodic and speaking style. These features involve the fundamental frequency (f0) and energy trajectories that can characterize speaker's identity. The fundamental frequency is obtained by the pitch tracking method, which as a function of time within a spoken utterance determines whether the speech is voiced or unvoiced and if it is voiced calculates f0.

Different approaches to speech analysis/synthesis naturally lead to different methods for pitch and voicing estimation. It is difficult to empirically measure the performance of an f0 estimator for several reasons, firstly performance depends on domain. Secondly, it is difficult to automatically rate the result of f0 estimator against expected outcomes, as it is difficult to measure f0 in the first place. In **[36]** a comparative performance study of pitch detection algorithms exists. Some of them are mentioned here:

- Time Domain method
    - Time-event Rate
        - ZRC(Zero crossing)
        - Peak Rate
        - Slop Event rate
    - Autocorrelation
    - The YIN estimator
- Frequency domain method
    - Component frequency Ratios
    - Filter-Based Method
    - Cepstrum Analysis
    - Multi-Resolution Method
- Statistical method
    - Neural Network
    - Maximum Likelihood Estimators

The Cepstrum Analysis method is implemented, where the Fourier transformation of the log of the magnitude spectrum of the input waveform was taken, which makes the nonlinear (inharmonic) system more linear. A frame-level F0 vector is generated with no post-processing steps, except from median filtering, which is essential in order to acquire better results for the rates of halving or doubling (due to estimator errors, often results give). Median filter is the average over ***k*** neighboring elements. In this specific thesis halving and doubling errors do not constitute a major problem as we fit all these features in a normal distribution. Thus if a small number of halvings or doublings exist in the vector, they represent a speaker's characteristic too. Meanwhile averaging all the values of F0 vector decreases the difference between speakers. Consequently, we decided on ***k=4*** for our experiments and indeed performed better than a smaller or a bigger window. Figure 3.9, shows the effect of median filtering on a speaker change utterance (applied during experiments).


***Voice signal***

***F0 Vector on a speaker change***

***BIC Performance***

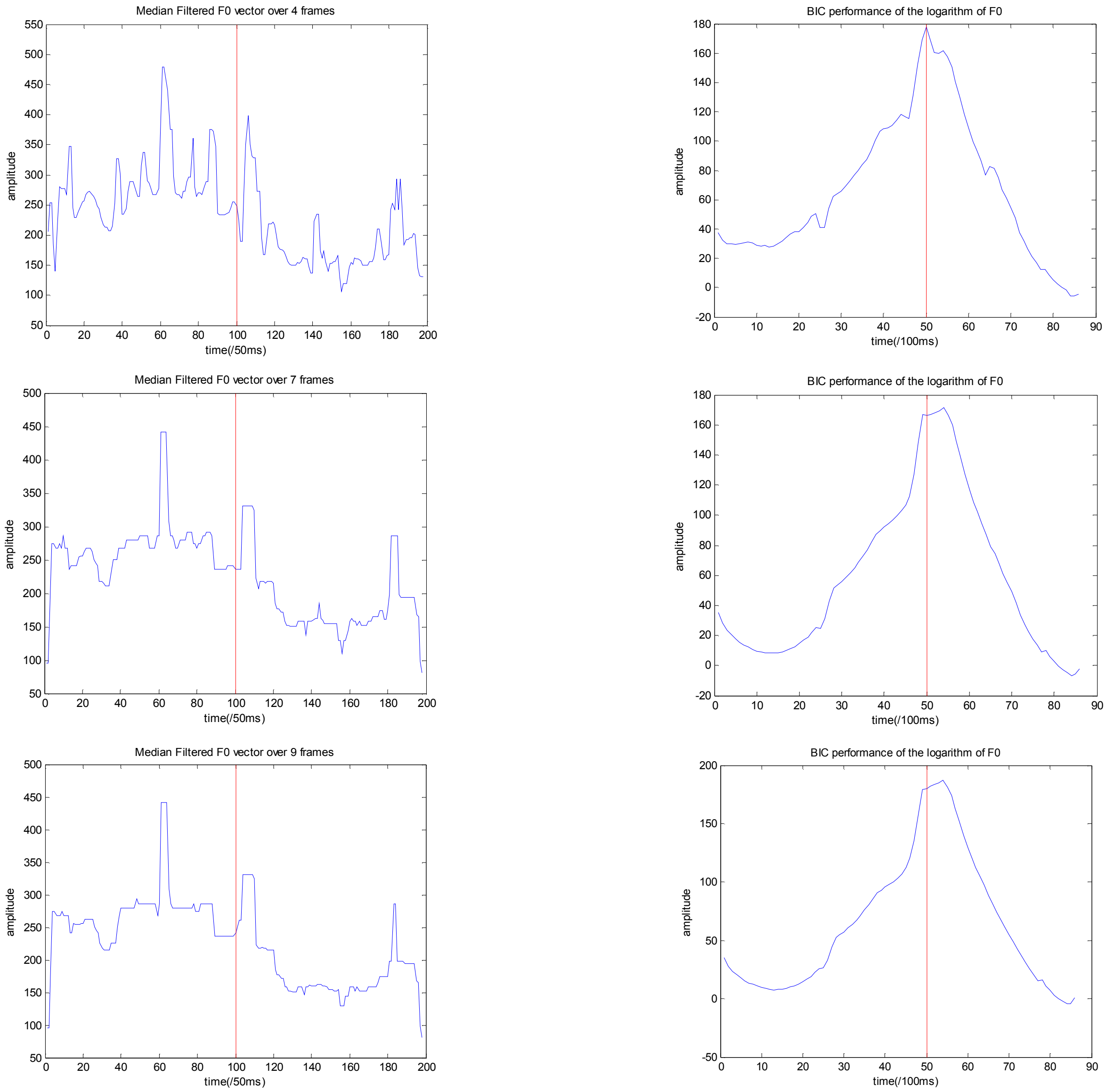


**Figure 3.10** On the left column median filtered F0 vectors are represented from four to nine elements of evidence. On the right column BIC performance for each filtered vector is shown.

The logarithm of (f0) unaccompanied was initially selected to represent prosody features, which although only a one-dimensional feature, it modeled audio stream extremely well and tasks continued to test the log (f0') (1st order derivative), energy and log (energy). The first-order derivative was estimated over a 5-frame context.

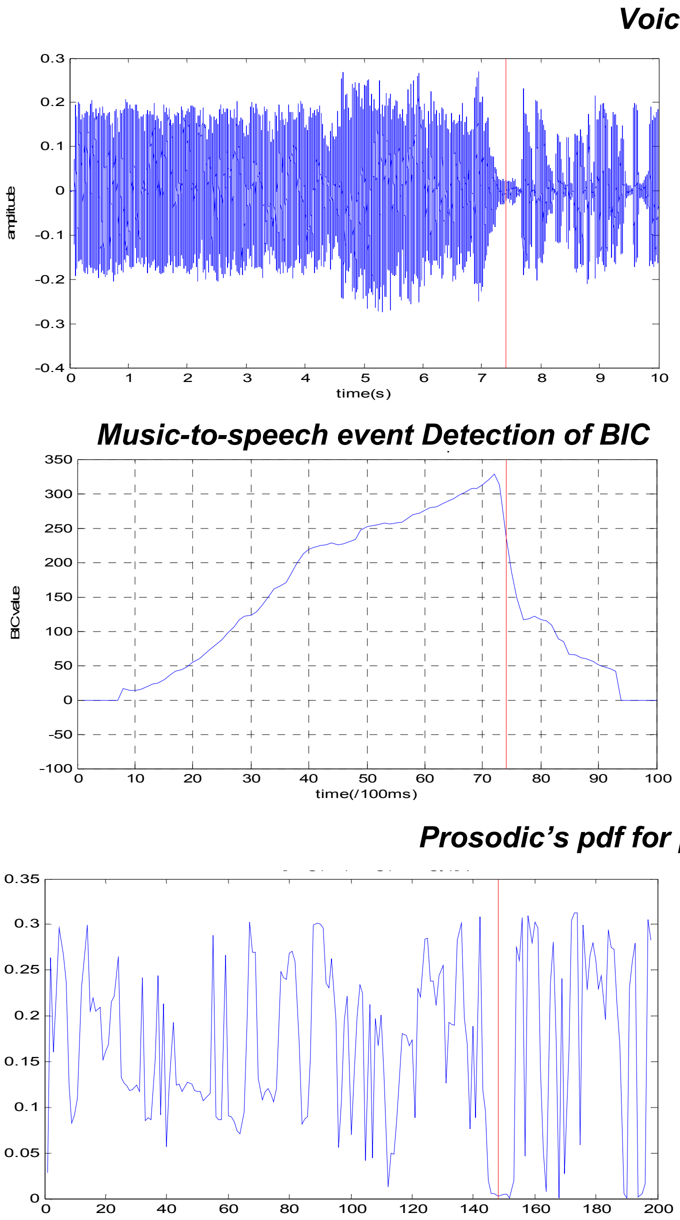


**Figure 3.10** On the first image the audio signal of a transition from music to speech appears (at 7.4 sec). Then the BIC values every 100ms for these 10 seconds are presented (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix).This estimation fell out by 100ms and is not considered as a false alarm. Finally the modeling of the signal by the prosodics is depicted.

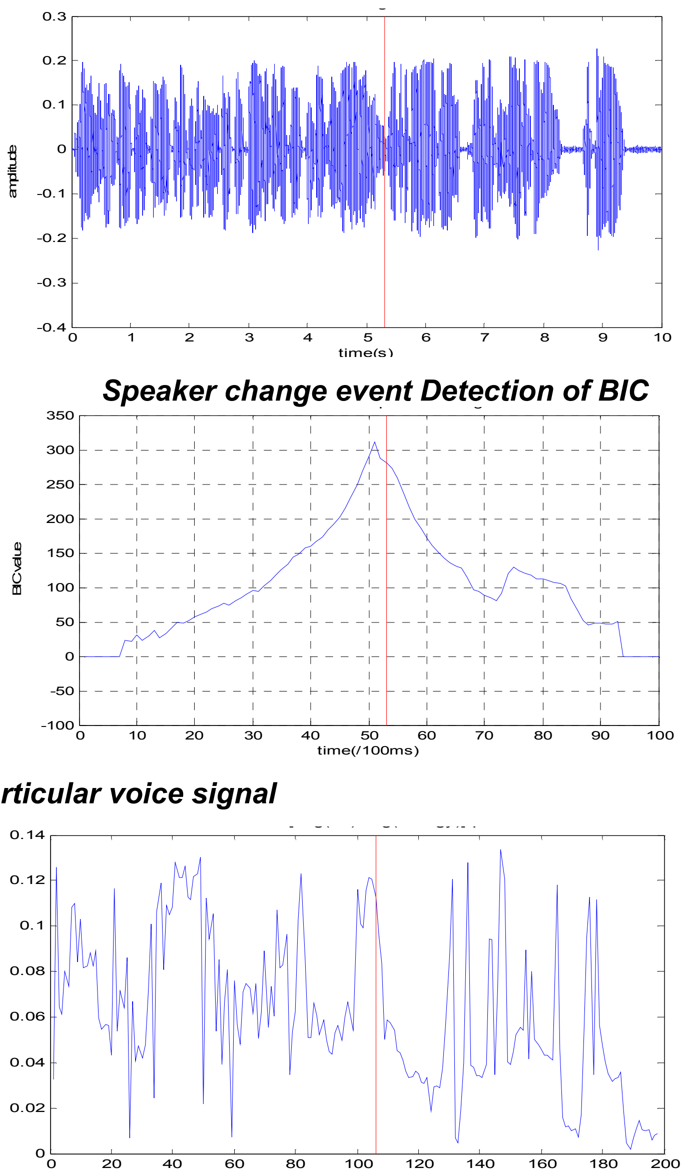


**Figure 3.11** On the first image the audio signal of a speaker change appears (at 5.3 sec). Then the BIC values every 100ms for these 10 seconds are presented (excluding 700 ms at start and 700ms at the end needed to estimate the determinant of the covariance matrix). This estimation fell out by 200ms and is not considered as a false alarm. Finally the modeling of the signal by the prosodics is depicted.

## 3.3 Corpus-Evaluation Measure and Baseline system Results

Now we are ready to evaluate our system beginning from a fixed frame length of baseline features and threshold in order to determine some of the parameters of BIC.

A commonly used figure to evaluate our results is the precision and recall defined as :

$$Precision = \frac{t_p}{f_p+t_p},$$

where $f_p$ denotes the false alarms and $t_p$ the correctly found segments. It is a function of true positives and examples misclassified as positives (false positives). It can show the quantity of the correct experiment results.

$$Recall = \frac{t_p}{f_n+t_p},$$

where $f_n$ denotes missed events. It is a function of its correctly classified examples (true positives) and its misclassified examples (false negatives).It shows the quality of the implementation. A function to determine the performance of a system using Recall and Precision is the F-measure.

$$F_{measure} = \frac{(\beta^2+1) * Precision * Recall}{(\beta^2 * \mathrm{Precision}) + Recall}$$

The F-measure is evenly balanced when β = 1. It favors precision when β> 1, and recall otherwise.

A false alarm occurs when a speaker turn is detected although it does not exist, a missed event occurs when we don’t detect an existing speaker turn.

The “Greek Broadcast” corpus we used has about 10 hours of various transcribed broadcast news shows recorded from April to May 2007. These data were obtained from NET and MEGA shows and were manually hand-segmented. Our audio data have the following characteristics:

- 1411 kbps bit rate
- 44 kHz audio sample rate
- 2 channel (stereo) one channel used in our experiments

The shows were selected specifically, so as to include all possible cases that were mentioned in **2.1** and are full length, including many commercials of

8:27 minutes length, which produce many false alarms. Results from ten broadcast news are represented below (five NET and five MEGA shows) length of 9:40:07

## *BASELINE SYSTEM RESULTS*

In the previous section the first set-up of the BIC criterion was described and now its results will be shown along with baseline features (MFCC's). More experiments are presented testing BICframe, the increase in step of the window size when there is no change detection. Through this step we concluded to some of the parameters that need to be determined in order to get a clear comparison of the features that will be used next and their length.

| frame = 100_50ms<br>THRESHOLD = 202 | BASELINE SYSTEM RESULTS<br>TABLE 3.1 | | |
|---|---|---|---|
| INC_Step=2sec | BICFRAME COMPARISON | | |
| | 8sec | 10sec | 12sec |
| Real Changes | 2253 | 2253 | 2253 |
| Matlab found[1] | 3146 | 2998 | 2811 |
| Correctly Found | 1988 | 1975 | 1908 |
| Missed | 265 | 278 | 345 |
| FA | 875 | 762 | 619 |
| Det0_120 | 1036 | 1640 > 75,78% | 978 |
| Det120_240 | 742 | 297 > 13,72% | 704 |
| Det240_360 | 243 | 115 > 5,31% | 237 |
| Det360_480 | 87 | 77 > 3,55% | 86 |
| Det_480_600 | 104 | 35 > 1,61% | 90 |
| MDR | 11,762% | 12,339% | 15,312% |
| FAR | 30,562% | 27,840% | 24,495% |
| PRC | 69,438% | 72,159% | 75,505% |
| RCL | 88,238% | 87,66% | 84,687% |
| F-measure | 77,717% | 79,157% | 79,832% |

***Det0_120*: segments detected in less or equal than 120 milliseconds. Following variables in time specified respectively, *MDR*: Miss Detection Rate, *FAR*: False Alarm Rate, *RCL*: Recall, *PRC*: Precision. [1]FA + correctly found should sum to Matlab found but we have handsegmented some locations twice due to decision problem, nevertheless we don't need-count both of these events.

We can see from the above table, that there is improvement in our system’s performance as BICFRAME increases but as expected there is bigger loss in event detections. In detail we get a 3,4% decrease in False Alarm rate while Miss Detection Rate increases by 3%.

| frame = 100_50ms<br>THRESHOLD = 202<br>INC_Step = 4sec | BASELINE SYSTEM RESULTS<br>TABLE 3.2 | | |
|---|---|---|---|
| | BICFRAME COMPARISON | | |
| | 8sec | 10sec | 12sec |
| Real Changes | 2253 | 2253 | 2253 |
| Matlab found | 2961 | 2805 | 2685 |
| Correctly Found | 1932 | 1937 | 1872 |
| Missed | 321 | 316 | 381 |
| FA | 746 | 585 | 530 |
| Det0_120 | 1010 | 978 | 956 |
| Det120_240 | 685 | 717 | 671 |
| Det240_360 | 258 | 228 | 247 |
| Det360_480 | 78 | 104 | 76 |
| Det_480_600 | 101 | 105 | 93 |
| MDR | 14,247% | 14,025% | 16,910% |
| FAR | 27,856% | 23,195% | 22,064% |
| PRC | 72,144% | 76,805% | 77,936% |
| RCL | 85,753% | 85,975% | 83,09% |
| F-measure | 78,362% | 81,131% | 80,430% |

The above experiment shows that as we increase the INC_Step, to expand the under examination BICFRAME when there is no change event, performance improves but at most in the module with BICFRAME=10seconds. We observe that the more data available in the criterion the better the results. However, miss detections increase because we risk the detection of one or more possible events. From now on in our modules we will maintain the BICFRAME=10 seconds and INC_Step=2 seconds.

## 3.4 Averaging and CMN to improve the performance

Being almost ready to present the results and the comparison of all previously mentioned feature utilizations on BIC segmentation method, some simple techniques were tested in order to improve overall performance. Firstly, we applied the averaging technique on MFCCs where we extract again ten-dimension coefficients every 20/15ms frame length. Every three sequential frames we produced a 50ms frame estimating the average value of them. Essentially we tried to include extra information on our feature vector, while on the same time avoiding over-segmentation due to small frame lengths, which increase considerably the system's sensitivity. Results showed improved detectability, however improved the insertion error too. We will see in detail the results in the following section using Equal Error Rate as a trustworthy comparison measure.

Sometimes, in speech recognition systems the characteristics of the channel may vary from one session to next, something that complicates the recognizer performance. Thus, Cepstral Mean Normalization (CMN) is applied in order to minimize the effect of these channel differences (in our case background noise). Essentially CMN involves subtracting the cepstral mean, calculated across the utterance, from each frame. In our implementation we apply the BIC criterion beginning with a ten second window, so we extract from this part our features and estimate their mean value, then this value is subtracted from each frame and gives a smoother distribution of our features. CMN surely improved the BIC performance.

# Chapter 4

## Experimental Results

Our implementation depended on too many variable parameters as explained further on. Our experiment compared 12, 10 and 8 second windows as well as 2 and 4 s increment steps, as explained in **3.3**. The most stable results were reached at 10 s with a 2 s increment step and therefore we decided to maintain these parameters throughout our experimental series.

### 4.1 Frame Comparison Results and Deductions

In the previous chapter several modules were tested, to decide which BICFRAME and increase step to choose for our implementation. At this stage we are seeking the most appropriate frame length, for the exploitation of the features, while maintaining a steady threshold.

# *Experimental Results — MFCC's*

| BICFRAME = 10sec<br>THRESHOLD = 202 | Mfcc's feature frame comparison<br>TABLE 4.1 | | |
|---|---|---|---|
| | 100_50ms | 50ms | 40ms |
| **Real Changes** | 2253 | 2253 | 2253 |
| **Matlab found** | 2998 | 2540 | 2747 |
| **Correctly Found** | 1975 | 1832 | 1855 |
| **Missed** | 278 | 421 | 398 |
| **FA** | 762 | 424 | 631 |
| **Det0_120** | 1640 > 75,78% | 1357 > 69,44% | 182 > 8,98% |
| **Det120_240** | 297 > 13,72% | 312 > 15,96% | 354 > 17,47% |
| **Det240_360** | 115 > 5,31% | 143 > 7,31% | 825 > 40,72% |
| **Det360_480** | 77 > 3,55% | 102 > 5,22% | 524 > 25,86% |
| **Det_480_600** | 35 > 1,61% | 40 > 2,04% | 141 > 6,95% |
| **MDR** | 12,339% | 18,686% | 17,665% |
| **FAR** | 27,840% | 18,794% | 25,383% |
| **PRC** | 72,159% | 81,205% | 74,617% |
| **RCL** | 87,66% | 81,313% | 81,502% |
| **F-measure** | 79,157% | 81,258% | 77,907% |

***Det0_120*: segments detected in less or equal than 120 milliseconds. Following variables in time specified respectively, *MDR*: Miss Detection Rate, *FAR*: False Alarm Rate, *RCL*: Recall, *PRC*: Precision.

# *Experimental Results — PLP’s*

| BICFRAME = 10sec<br>THRESHOLD = 202 | PLP’s feature frame comparison<br>TABLE 4.2 | | | | | |
|---|---|---|---|---|---|---|
| | 100_50 ms frame | | 50 ms frame | | 40 ms frame | |
| **Real Changes** | 2253 | | 2253 | | 2253 | |
| **Matlab found** | 3377 | | 2726 | | 3056 | |
| **Correctly Found** | 1911 | | 1811 | | 1825 | |
| **Missed** | 342 | | 442 | | 428 | |
| **FA** | 1205 | | 754 | | 970 | |
| **Det0_120** | 1405 | 66,49% | 1281 | 63,66% | 216 | 10,74% |
| **Det120_240** | 394 | 18,64% | 358 | 17,79% | 408 | 20,28% |
| **Det240_360** | 144 | 6,81% | 191 | 9,49% | 814 | 40,47% |
| **Det360_480** | 135 | 6,38% | 111 | 5,51% | 454 | 22,57% |
| **Det_480_600** | 52 | 2,46% | 61 | 3,03% | 119 | 5,91% |
| **MDR** | 15,179% | | 19,618% | | 18,996% | |
| **FAR** | 38,672% | | 29,396% | | 34,705% | |
| **PRC** | 61,328% | | 70,604% | | 65,295% | |
| **RCI** | 84,82% | | 80,381% | | 81,003% | |
| **F-measure** | 71,185% | | 75,175% | | 72,305% | |

PLP’s respond well to Bayesian Information Criterion achieving more than 75% success.

## *Observations obtained by comparing the length of the frame.*

1. A longer frame resulted in greater accuracy in hand segmented intervals as compared to the MATLAB program.
2. Frames with overlaps lead to a marked increase in false alarms.
3. In frames with no overlap we noted that at 50 ms the false alarm rate was lower than at 40 ms, with a slight increase in the miss detection rate.

If false alarms were not taken into account then a small frame with overlap would have been chosen in order to maximize the detectability of our system.

### 4.2 Feature specific set-ups and Results

In this section we describe the parameters and the modules implemented for each feature included in our experiments aside from the MFCC's and PLP's. Through this section we decide the best set-ups for every feature.

## *RASTA_PLP SET-UP and RESULTS*

We introduced the averaging technique in order to improve the performance of the RASTA PLPs. In doing this, the results were comparable to MFCCs and PLPs and a yield of 70 % was obtained. Threshold and BICFRAME maintained as previously.

| BICFRAME=10sec<br>THRESHOLD = 202 | RASTA_PLP’s with averaging mode<br>TABLE 4.3<br>40 ms |
|---|---|
| Real Changes | 2253 |
| Matlab found | 2894 |
| Correctly Found | 1701 |
| Missed | 552 |
| Misses btn 600_1000 | 136 |
| FA | 932 |
| Det0_120 | 488 26,36% |
| Det120_240 | 606 32,74% |
| Det240_360 | 391 21,12% |
| Det360_480 | 206 11,12% |
| Det_480_600 | 160 8,64% |
| MDR | 24,5% |
| FAR | 35,396% |
| PRC | 64,603% |
| RCL | 75,499% |
| F-measure | 69,627% |

**Table 4.3** shows that Rasta_PLP’s reduced the performance comparatively to previously used features. Specifically they gave worse MDR and FAR, while they were expected to produce a small insertion rate.

# *SDCC’s SET-UP and RESULTS*

Short frame lengths produce high false alarm rates while SDCCs require short frames as a result of the high dimensional feature vectors. We implemented the SDCC parameters N=10, d=1, P=3, k=2. This means *10(mfcc’s)*2(concatenated deltas) = 20* characteristics for every frame.

| **BICFRAME = 10sec**<br>**THRESHOLD = -280** | SDCC’s Results<br>TABLE 4.4.1 |
|---|---|
| | 30 ms N,d,P,k parameters 10,1,3,2 |
| **Real Changes** | 2253 |
| **Matlab found** | 3011 |
| **Correctly Found** | 1722 |
| **Missed** | 531 |
| **Misses btn 600_1000** | 128 |
| **FA** | 1006 |
| **Det0_120** | 769 |
| **Det120_240** | 590 |
| **Det240_360** | 246 |
| **Det360_480** | 162 |
| **Det480_600** | 113 |
| **MDR** | 23,568% |
| **FAR** | 36,876% |
| **PRC** | 63,124% |
| **RCL** | 76,431% |
| **F-measure** | 69,143% |

Testing bigger frames than on the above setup, we observed worse results.

| BICFRAME = 10sec<br>THRESHOLD = -280 | SDCC's Results different setup<br>TABLE 4.4.2 |
|---|---|
| | 30 ms N,d,P,k parameters 10,1,3,3 |
| Real Changes | 2253 |
| Matlab found | 3183 |
| Correctly Found | 1691 |
| Missed | 562 |
| Misses btn 600_1000 | 137 |
| FA | 1209 |
| Det0_120 | 816 |
| Det120_240 | 421 |
| Det240_360 | 243 |
| Det360_480 | 181 |
| Det480_600 | 144 |
| MDR | 24,944% |
| FAR | 41,689% |
| PRC | 58,310% |
| RCL | 75,055% |
| F-measure | 65,631% |

At the beginning of our experiment we noted the BICFRAME should be as small as possible but note that for SDCCs, we need a larger BICFRAME because of the high dimensionality they require.

# *Prosodic SET-UP and RESULTS*

A frame of 100/50 ms was finally selected for the pitch tracking because smaller ones were producing high false alarm rate. We quote the results for the Prosodic features beginning from one dimensional feature and then increasing dimensions on them .Adding energy and the 1st order derivative with no independence at least in principle.

| **BICFRAME = 10sec**<br>**THRESHOLD = 45/65** | Prosodics Results log(F0)<br>TABLE 4.5.1<br>100/50 ms | |
|---|---|---|
| **Real Changes** | 2253 | |
| **Matlab found** | 3149 | 2218 |
| **Correctly Found** | 1074 | 973 |
| **Missed** | 1179 | 1280 |
| **Misses btn 600_1000** | 141 | 102 |
| **FA** | 1781 | 951 |
| **MDR** | 52,33% | 56,813% |
| **FAR** | 62,381% | 49,428% |
| **PRC** | 47,67% | 50,572% |
| **RCL** | 37,619% | 43,187% |
| **F-measure** | 42,052% | 46,588% |

A one dimensional feature on BIC giving 46,588% rate of success seems very promising and we could use it in combination with other acoustic features.

| BICFRAME = 10sec<br>THRESHOLD = 75 | Prosodics Results [log(F0) log(der(F0))]<br>TABLE 4.5.2<br>100/50 ms |
|---|---|
| Real Changes | 2253 |
| Matlab found | 2416 |
| Correctly Found | 972 |
| Missed | 1281 |
| Misses btn 600_1000 | 110 |
| FA | 1150 |
| MDR | 56,857% |
| FAR | 54,194% |
| PRC | 45,806% |
| RCL | 43,143% |
| F-measure | 44,434% |

Adding the first order derivative of pitch in the Prosodic features increased the Error rates. More specifically it increased the FAR by almost 5%. So it will be excluded from later experiments.

| BICFRAME = 10sec<br>THRESHOLD = 75/85 | Prosodics Results [log(F0) Energy]<br>TABLE 4.5.3<br>100/50 ms | |
|---|---|---|
| Real Changes | 2253 | |
| Matlab found | 2884 | 2626 |
| Correctly Found | 1172 | 1143 |
| Missed | 1081 | 1110 |
| Misses btn 600_1000 | 169 | 149 |
| FA | 1418 | 1189 |
| MDR | 47.98% | 49,267% |
| FAR | 54,749% | 50,986% |
| PRC | 45,251% | 49,014% |
| RCL | 52,02% | 50,733% |
| F-measure | 48,399% | 49,858% |

On the other hand Energy seems to be a valuable addition, giving a rise of success rate by 5.4%.

| BICFRAME = 10sec<br>THRESHOLD = 85 | Prosodics Results [log(F0) log(Energy)]<br>TABLE 4.5.4<br>100/50 ms |
|---|---|
| Real Changes | 2253 |
| Matlab found | 2911 |
| Correctly Found | 1271 |
| Missed | 982 |
| Misses btn 600_1000 | 155 |
| FA | 1346 |
| MDR | 43,586% |
| FAR | 51,432% |
| PRC | 48,568% |
| RCL | 56,414% |
| F-measure | 52,197% |

Exploiting just the logarithm of energy of the signal improves the results by 2.3%. It appears to be distributed more in common with pitch.

| BICFRAME = 10sec<br>THRESHOLD = 110 | Prosodics Results [log(F0) log(Energy) der(Energy)]<br>TABLE 4.5.5<br>100/50 ms |
|---|---|
| Real Changes | 2253 |
| Matlab found | 2912 |
| Correctly Found | 1269 |
| Missed | 984 |
| Misses btn 600_1000 | 169 |
| FA | 1349 |
| MDR | 43,675% |
| FAR | 51,527% |
| PRC | 48,473% |
| RCL | 56,325% |
| F-measure | 52,104% |

A slight increase in Miss Detection Rate is observed including the Energy derivative. We proceed without utilizing it in the following experiments.

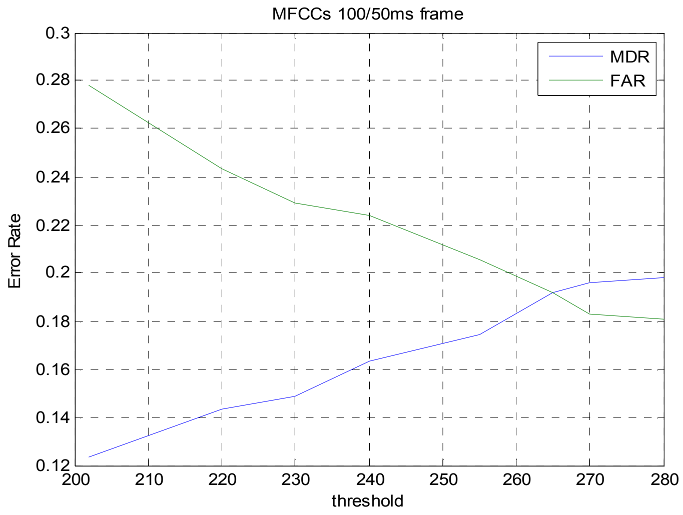
Unfortunately absolute comparison cannot be attained because there are two kinds of errors and the most reliable one would be the equal error rate. Thus, we continue to search for that threshold at which equal error of both kinds can be produced in every module.

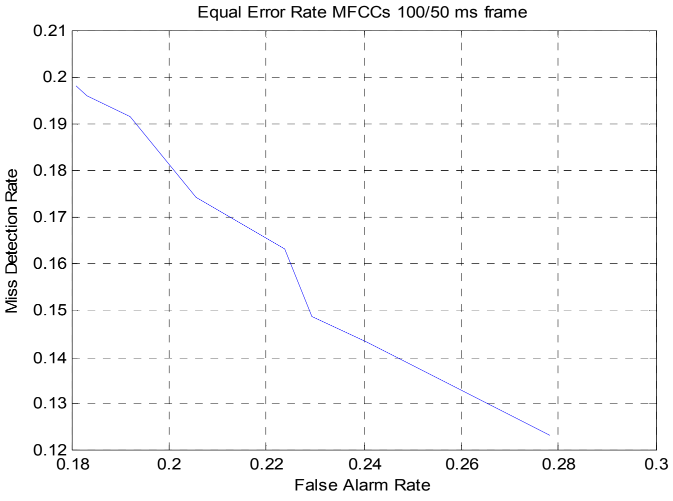
## 4.3 Equal Error Rate Results-Feature Comparison

What changes in the following experiments is the threshold. It will produce for every “ideal” set-up until now an equal error rate which will help us decide which features are the most appropriate for the criterion.

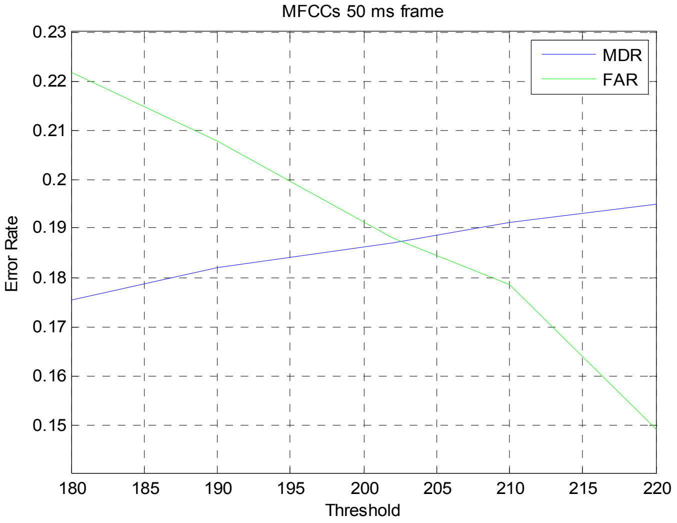
### *Mfcc’s EER*

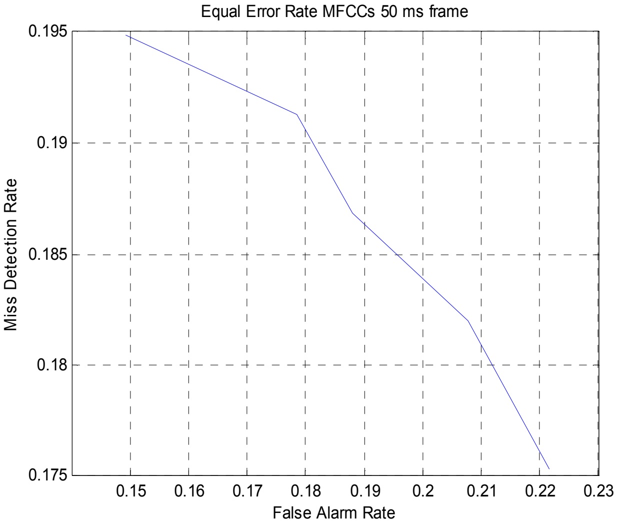
| MDR_FAR | MFCC’s Equal Error Rate on 100/50ms frame<br>TABLE 4.6.1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| THRESHOLD | 202 | 220 | 230 | 240 | 255 | 265 | 270 | 280 |
| MDR | 12,339% | 14,336% | 14,869% | 16,333% | 17,443% | 19,174% | 19,618% | 19,84% |
| FAR | 27,84% | 24,343% | 22,94% | 22,396% | 20,546% | 19,21% | 18,313% | 18,095% |
| Missed | 278 | 323 | 335 | 368 | 393 | 432 | 442 | 447 |
| FA | 762 | 612 | 571 | 544 | 481 | 433 | 406 | 399 |



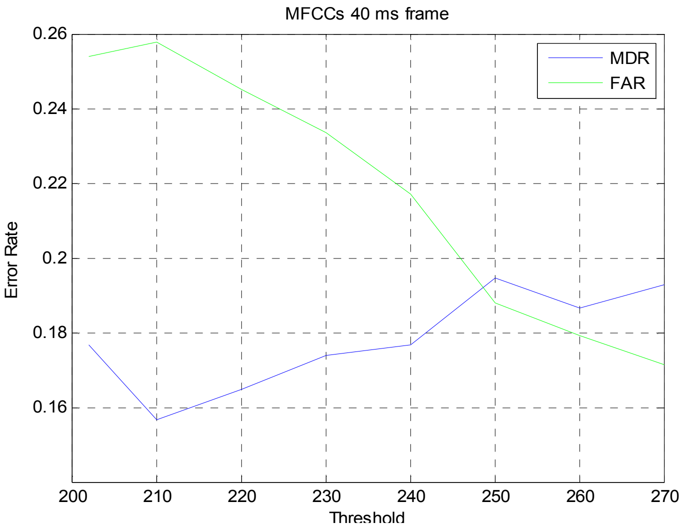


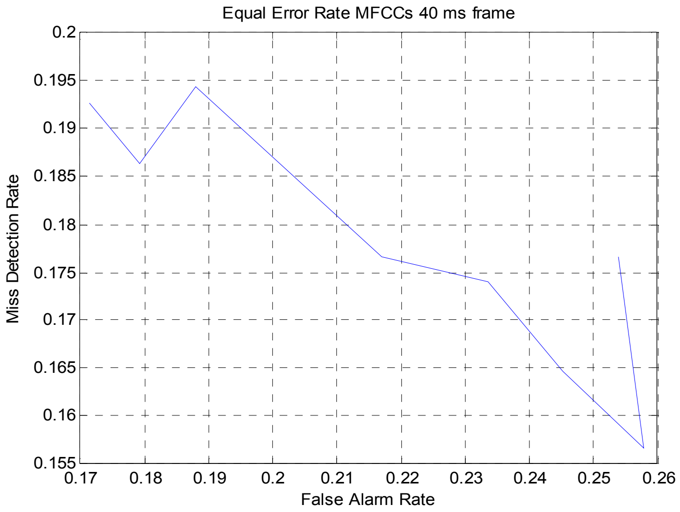
| MDR_FAR | MFCC's Equal Error Rate on 50ms frame<br>TABLE 4.6.2 | | | | |
|---|---|---|---|---|---|
| THRESHOLD | 180 | 190 | 202 | 210 | 220 |
| MDR | 17,532% | 18,197% | 18,686% | 19,13% | 19,485% |
| FAR | 22,161% | 20,765% | 18,794% | 17,853% | 14,915% |
| Missed | 395 | 410 | 421 | 431 | 439 |
| FA | 529 | 483 | 424 | 396 | 318 |





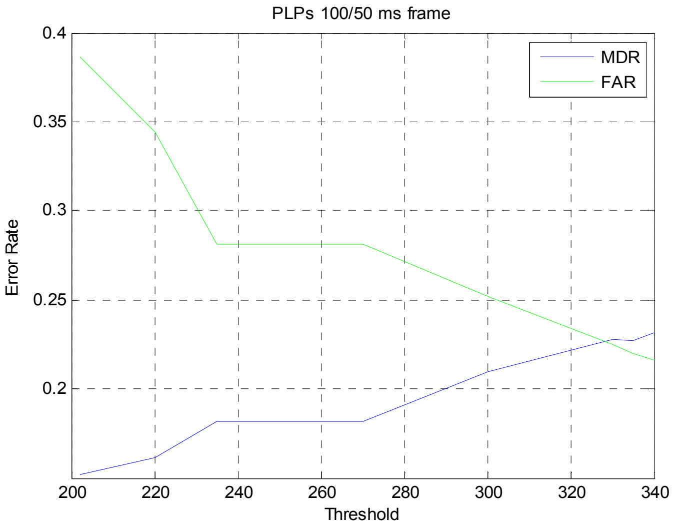
| MDR_FAR | MFCC's Equal Error Rate on 40ms frame<br>TABLE 4.6.3 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| THRESHOLD | 202 | 210 | 220 | 230 | 240 | 250 | 260 | 270 |
| MDR | 17,665 % | 15,667 % | 16,466 % | 17,399 % | 17,665 % | 19,44 % | 18,641 % | 19,263 % |
| FAR | 25,38 % | 25,781 % | 24,508 % | 23,352 % | 21,696 % | 18,791 % | 17,913 % | 17,129 % |
| Missed | 398 | 353 | 371 | 392 | 398 | 438 | 420 | 434 |
| FA | 631 | 660 | 611 | 567 | 514 | 420 | 400 | 376 |

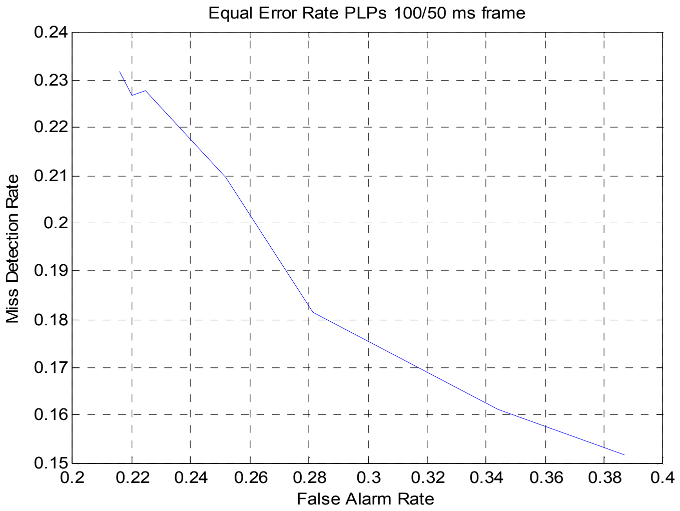




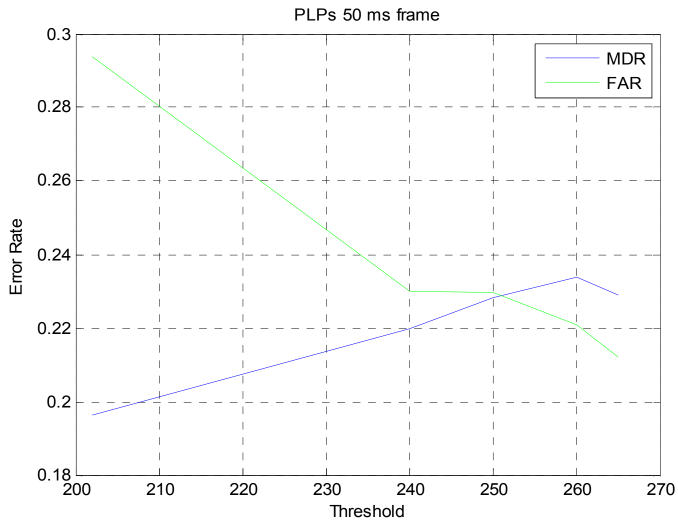
# *PLPs EER*

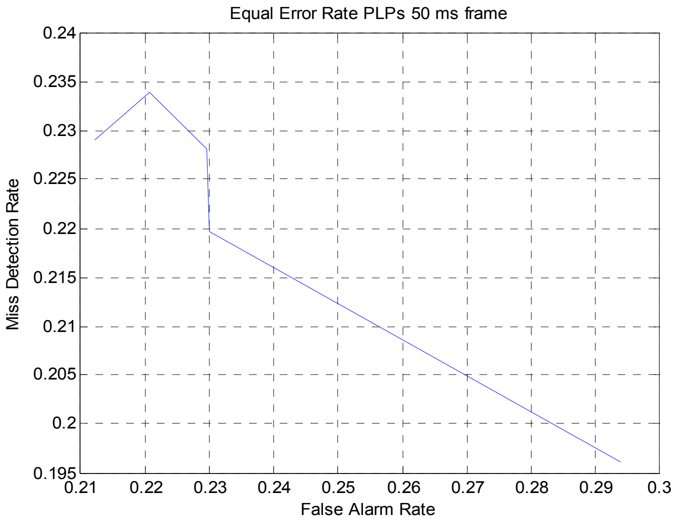
| MDR_FAR | PLP's Equal Error Rate on 100/50ms frame<br>TABLE 4.7.1 | | | | | | |
|---|---|---|---|---|---|---|---|
| THRESHOLD | 202 | 220 | 235-270 | 300 | 330 | 335 | 340 |
| MDR | 15,17% | 16,11% | 18,15% | 20,94% | 22,76% | 22,68% | 23,16% |
| FAR | 38,67% | 34,42% | 28,13% | 25,19% | 22,45% | 21,98% | 21,60% |
| Missed | 342 | 363 | 409 | 472 | 513 | 511 | 522 |
| FA | 1205 | 992 | 722 | 600 | 504 | 491 | 477 |



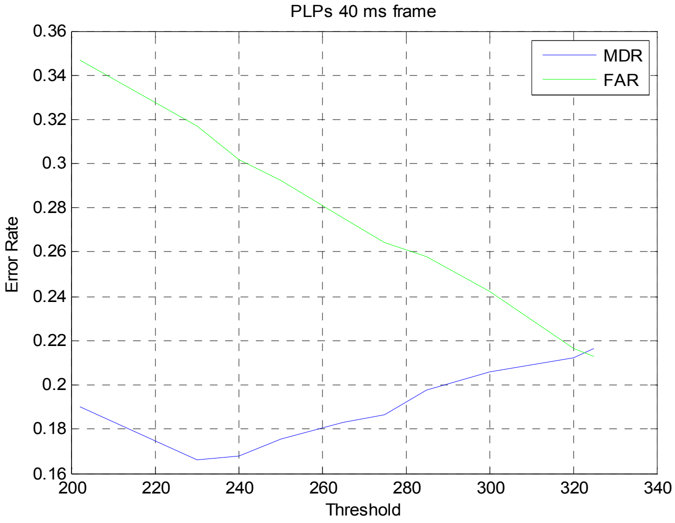


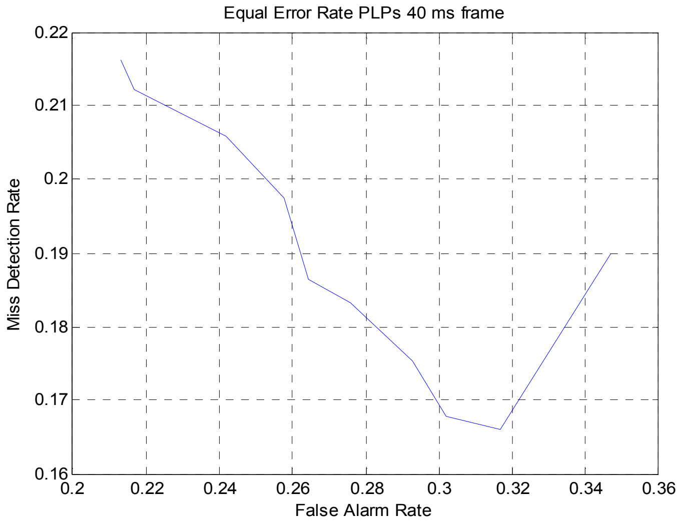
| MDR_FAR | PLP's Equal Error Rate on 50ms frame<br>TABLE 4.7.2 | | | | |
|---|---|---|---|---|---|
| THRESHOLD | 202 | 240 | 250 | 260 | 265 |
| MDR | 19,618% | 21,97% | 22,814% | 23,391% | 22,902% |
| FAR | 29,395% | 22,996% | 22,95% | 22,076% | 21,22% |
| Missed | 442 | 495 | 514 | 527 | 516 |
| FA | 754 | 525 | 518 | 489 | 468 |





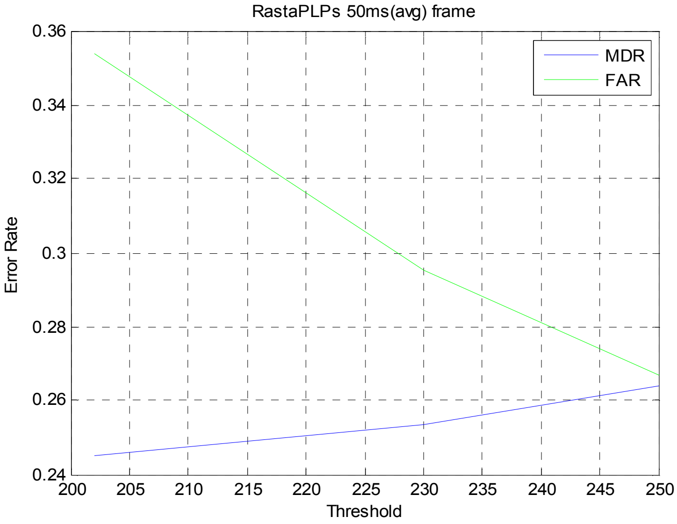
| MDR_FAR | PLP's Equal Error Rate on 40ms frame<br>TABLE 4.7.3 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| THRESHOLD | 202 | 230 | 240 | 250 | 265 | 275 | 285 | 300 | 320 | 325 |
| MDR | 18,99% | 16,6% | 16,77% | 17,53% | 18,33% | 18,64% | 19,75% | 20,59% | 21,21% | 21,61% |
| FAR | 34,70% | 31,67% | 30,19% | 29,27% | 27,55% | 26,41% | 25,77% | 24,19% | 21,66% | 21,3% |
| Missed | 428 | 374 | 378 | 395 | 413 | 420 | 445 | 464 | 478 | 487 |
| FA | 970 | 871 | 811 | 769 | 700 | 658 | 628 | 571 | 491 | 478 |

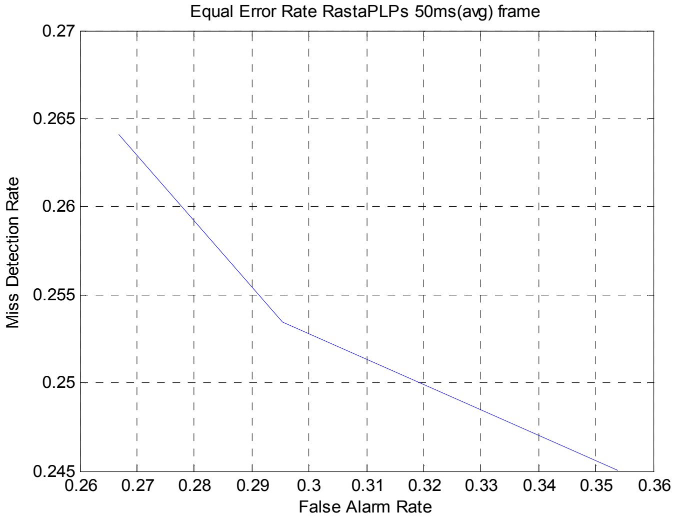




# *RASTA_PLPs EER*

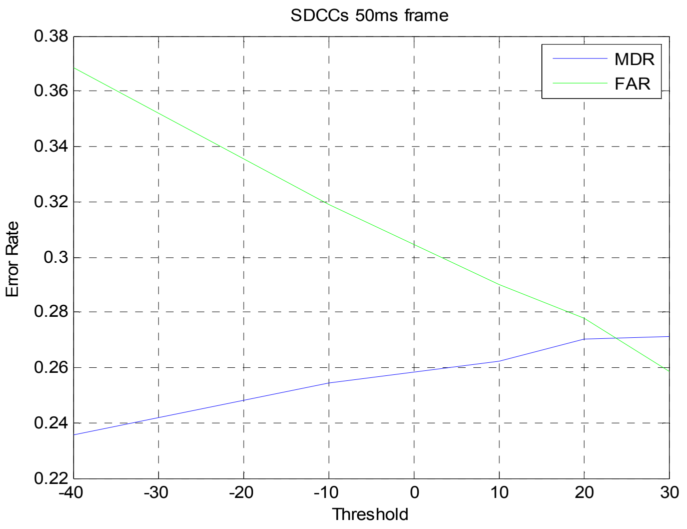
| MDR_FAR | RASTA_PLP's Equal Error Rate on 40ms(avg) frame<br>TABLE 4.8 | | |
|---|---|---|---|
| THRESHOLD | 202 | 230 | 250 |
| MDR | 24,5% | 25,343% | 26,409% |
| FAR | 35,396% | 29,534% | 26,669% |
| Missed | 552 | 571 | 595 |
| FA | 932 | 705 | 603 |

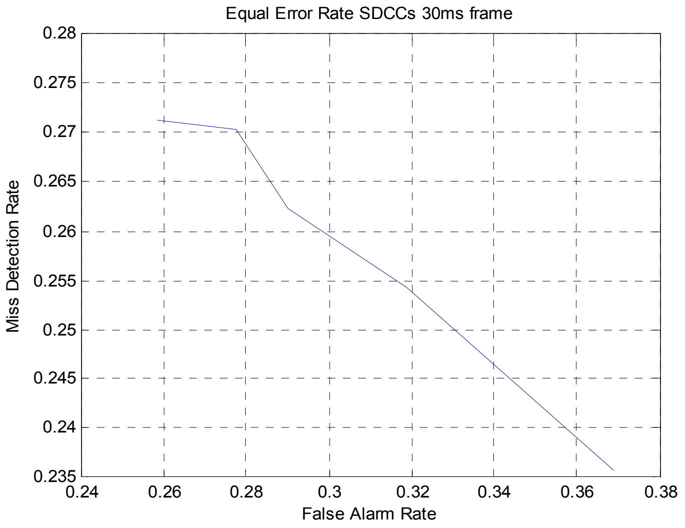




# SDCC’s EER

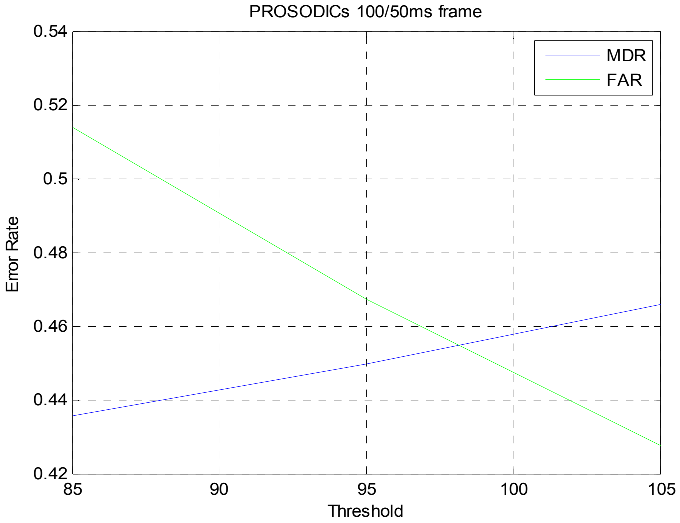
| MDR_FAR | SDCC’s Equal Error Rate on 30ms(10,1,3,2) frame<br>TABLE 4.9 | | | | |
|---|---|---|---|---|---|
| **THRESHOLD** | -40 | -10 | 10 | 20 | 30 |
| **MDR** | 23,568% | 25,432% | 26,231% | 27,03% | 27,119% |
| **FAR** | 36,876% | 31,873% | 29,004% | 27,768% | 25,869% |
| **Missed** | 531 | 573 | 591 | 609 | 611 |
| **FA** | 1006 | 786 | 679 | 632 | 573 |

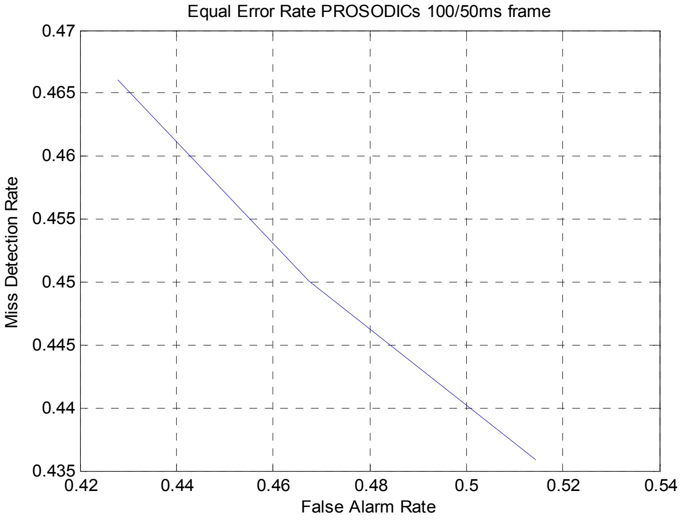




# *PROSODIC's EER*

| MDR_FAR | PROSODIC's EER 100/50ms log(F0)<br>TABLE 4.10 | | |
|---|---|---|---|
| **THRESHOLD** | 85 | 95 | 105 |
| **MDR** | 43,586% | 45,006% | 46,604% |
| **FAR** | 51,432% | 46,755% | 41,802% |
| **Missed** | 982 | 1014 | 1050 |
| **FA** | 1346 | 1088 | 900 |





# *Experimental Results – Comments and Comparisons.*

1. Threshold tuning is necessary every time feature frame, feature type and generally time-dependent parameters such as the step at which we search change detection changes.
2. The EER based comparison, shows that the best results were obtained by MFCCs on a forty millisecond frame, yielding an error rate of 18,35%.
3. Again a forty millisecond frame performed the best utilizing PLPs. The final error rate with these features was 21,45%.
4. RASTA_PLP's didn't respond well on this feature comparison. A result we expected since these particular characteristics are frequently used to normalize environmental effects on speech vectors. In using the average technique with a forty millisecond frame, we successfully reduced the error rate to 26,5%.
5. SDCCs didn't manage to correctly model and differentiate inhomogeneous acoustic vectors, inducing over-segmentation and a high miss detection rate. Through these experiments, it became obvious that multi-dimensional features do not contribute to BIC criterion. SDCCs best results were obtained with a frame of thirty milliseconds and a 27,4% error rate.
6. Although a one-dimensional characteristic, Prosodics induce a 45,9% error rate. Experiments showed that they can be combined with MFCCs resulting in a better performance.
7. It is worth mentioning that 45% of the total of false alarms were inserted from two out of ten broadcast news shows in our corpus. The increased difficulty resulting from the numerous commercials during these two particular newscasts, affected the final threshold on the equal error rate search. If we had changed our corpus, we would have decreased the threshold and improved the detectability, avoiding many insertions. However, we would have lost the opportunity to design a competitive and comprehensive automatic segmentation system.

Even though the forty millisecond frame seems to be the most productive, the need to vary the parameters required us to change this frame in the experiments that follow.

## 4.4 Implementing techniques and evaluating Results

In this section we will compare results that emerge from the averaging and the normalization techniques

### *Averaging*

On this 50ms MFCC's frame setup, we examine averaging using the **same BICFRAME, increase step and threshold**.

| BICFRAME=10sec<br>THRESHOLD = 202 | Mfcc's through averaging technique<br>TABLE 4.11<br>50 ms averaging 3*(20/15ms) |
|---|---|
| **Real Changes** | 2253 |
| **Matlab found** | 2990 |
| **Correctly Found** | 1928 |
| **Missed** | 325 |
| **Misses btn 600_1000** | 100 |
| **FA** | 801 |
| **Det0_120** | 885 |
| **Det120_240** | 789 |
| **Det240_360** | 237 |
| **Det360_480** | 106 |
| **Det480_600** | 93 |
| **MDR** | 14,425% |
| **FAR** | 29,351% |
| **PRC** | 70,648% |
| **RCL** | 85,574% |
| **F-measure** | 77,397% |

The corresponding table **4.1** reflecting the non-averaged 50 ms frame produced better results, at this first level of comparison on a particular threshold.

| BICFRAME=10sec<br>THRESHOLD = 202 | Mfcc’s through averaging technique<br>TABLE 4.12<br>40 ms averaging 3*(20/10s) |
|---|---|
| Real Changes | 2253 |
| Matlab found | 4009 |
| Correctly Found | 1953 |
| Missed | 300 |
| Misses btn 600_1000 | 119 |
| FA | 1737 |
| Det0_120 | 727 |
| Det120_240 | 939 |
| Det240_360 | 305 |
| Det360_480 | 133 |
| Det480_600 | 106 |
| MDR | 13,315% |
| FAR | 47,073% |
| PRC | 52,927% |
| RCL | 86,685% |
| F-measure | 65,724% |

The corresponding table **4.1** reflecting the non-averaged 40 ms frame produced far better results.

# *CEPSTRAL MEAN NORMALIZATION*

Again, we experiment with a previous setup and the same BICFRAME, increase step and threshold, including the normalization technique.

| BICFRAME = 10sec | MFCC's through CMN technique |
|---|---|
| THRESHOLD = 202 | TABLE 4.13 |
| | 100/50ms |
| **Real Changes** | 2253 |
| **Matlab found** | 3009 |
| **Correctly Found** | 1977 |
| **Missed** | 276 |
| **Misses 600_1000** | 89 |
| **FA** | 771 |
| **Det0_120** | 1413 |
| **Det120_240** | 442 |
| **Det240_360** | 172 |
| **Det360_480** | 70 |
| **Det480_600** | 71 |
| **MDR** | 12,25% |
| **FAR** | 25,496% |
| **PRC** | 74,503% |
| **RCL** | 87,749 |
| **F-measure** | 79,063% |

We observe a 0,089% increase in Recall and a 2,344% increase in Precision, utilizing the CMN method.

Then we repeated the test on a different frame to verify that normalization improves the results

| BICFRAME = 10sec<br>THRESHOLD = 202 | Mfcc's through CMN<br>TABLE 4.14 |
|---|---|
| | 50 ms |
| Real Changes | 2253 |
| Matlab found | 2521 |
| Correctly Found | 1845 |
| Missed | 408 |
| Misses btn 600_1000 | 61 |
| FA | 395 |
| Det0_120 | 477 |
| Det120_240 | 804 |
| Det240_360 | 489 |
| Det360_480 | 108 |
| Det480_600 | 120 |
| MDR | 18,109% |
| FAR | 17,633% |
| PRC | 82,366% |
| RCL | 81,890% |
| F-measure | 82,127% |

An increase in Precision (1,161%) results in an increase in recall (0,577%). Precision is positively influenced by implementing CMN.

And finally we include the normalization technique after the averaging one tested earlier to verify the results.

| BICFRAME = 10sec<br>THRESHOLD = 202 | MFCC's through averaging and CMN<br>TABLE 4.15 |
|---|---|
| | 50 ms 3*(20/15ms) |
| **Real Changes** | 2253 |
| **Matlab found** | 2990 |
| **Correctly Found** | 1948 |
| **Missed** | 305 |
| **Misses btn** | 88 |
| **FA** | 759 |
| **Det0_120** | 896 |
| **Det120_240** | 798 |
| **Det240_360** | 441 |
| **Det360_480** | 109 |
| **Det480_600** | 94 |
| **MDR** | 13,53% |
| **FAR** | 28,038% |
| **PRC** | 71,961% |
| **RCL** | 86,462% |
| **F-measure** | 78,547% |

Thus experimentally we have now proven that this simple technique (CMN) improves Precision (1,313%) and Recall (0,888%) . Essentially, CMN should be included in segmentation tasks guaranteeing a slight improvement in the system's performance

# Chapter 5

## FEATURE COMBINATION

Empirically, every parameter that may affect the results is taken into consideration, including the BICframe length and the step that we follow to estimate a BIC value, which is approximately 100ms (depending on frame length).A number of different acoustic features have been examined and tested on BIC and their performance recorded. Now we proceed to examining the problem of combining these features with the best setups that emerged.

The criterion will not have to change as far as distributions are concerned, again Gaussian distributions are assumed. The formula of BIC of the union of features is represented in this unit. The new criterion of BIC is as follows for the example of combining as independent the pitch and energy of the signal under examination. In chapter **3.1** the analysis of BIC was introduced stating:

$$R(i) = N\log|\Sigma| - N\log|\Sigma_1| - N\log|\Sigma|_2$$

It should be mentioned here that $R(i)$ essentially is the entropy of a Gaussian and the criterion chooses to split the vector of the observations when the sum of two smaller vectors maximizes the uncertainty.

$H(X,Y) = H(X) + H(Y|X)$ , H is the entropy of the Gaussian.

$$H(Y|X) \approx Independent$$

So

$$H(X,Y) = H(X) + H(Y)$$

Then

$BIC_x = NH(X) - \lambda_x P_x$ , N is the number of data.

$$BIC_Y = NH(Y) - \lambda_Y P_Y$$

$$BIC_{X,Y} = NH(X,Y) - \lambda_{XY} P_{X+Y}$$

$\lambda_{XY} = 1$ , λ is the penalty weight which we don't change.

$P_{X+Y} = P_X + P_Y$ , P is the dimension penalty.

And finally

$$BIC_{X,Y} = BIC_X + BIC_Y$$

This module produced the following results

| BICFRAME = 10sec<br>THRESHOLD = 85<br>100/50ms | Prosodics Independent<br>log(F0),log(Energy)<br>TABLE 5.1 |
|---|---|
| Real Changes | 2253 |
| Matlab found | 2758 |
| Correctly Found | 1209 |
| Missed | 1044 |
| Misses btn 600_1000 | 165 |
| FA | 1255 |
| MDR | 46,338% |
| FAR | 50,933% |
| PRC | 49,067% |
| RCL | 53,662% |
| F-measure | 51,261% |

Experiments showed that using the prosodics $\log F0$ *and* $\log Energy$ in combination (dependent) as per F-measure improved results by almos one percent better results for this setup. We conclude that Energy and pitch features together perform far better than when standing alone.

We proceed to experiment with a combination of features of MFCC's with Prosodics as above. Next we implement several frame lengths of Prosodics and Mfcc's at the same time. We note here that the method we implement to extract the Prosodic features require frames with overlaps. As shown below the frames tested are 100_50ms, 80_40ms, 60_30ms, and 40_20ms.

| **BICFRAME=10sec**<br>**100/50 ms** | Feature Combination [MFCC's],[Prosodics]<br>TABLE 5.2.1 | | | | | |
|---|---|---|---|---|---|---|
| **THRESHOLD** | 200 | 220 | 240 | 260 | 280 | 300 |
| **Real Changes** | 2253 | 2253 | 2253 | 2253 | 2253 | **2253** |
| **Matlab found** | 3046 | 2855 | 2810 | 2730 | 2638 | **2549** |
| **Correctly Found** | 1926 | 1958 | 1894 | 1869 | 1843 | **1823** |
| **Missed** | 327 | 295 | 359 | 384 | 410 | **430** |
| **Misses btn 600_1000** | 91 | 109 | 74 | 70 | 73 | **71** |
| **FA** | 836 | 780 | 632 | 658 | 489 | **429** |
| **MDR** | 14,513% | 13,093% | 15,934% | 17,043% | 18.197% | **19,085%** |
| **FAR** | 30,267% | 28,487% | 25,019% | 26,038% | 20.969% | **19,049%** |
| **PRC** | 69,733% | 71,513% | 74,981% | 73,962% | 79.031% | **80,951%** |
| **RCL** | 85,487% | 86,907% | 84,066% | 82,957% | 81,803% | **80,915%** |
| **F-measure** | 76,81% | 78,462% | 79,264% | 78,201% | 80,393% | **80,932%** |

| BICFRAME=10sec<br>80/40 ms | Feature Combination [MFCC’s],[Prosodics]<br>TABLE 5.2.2 | | |
|---|---|---|---|
| THRESHOLD | 300 | 330 | 360 |
| Real Changes | 2253 | 2253 | 2253 |
| Matlab found | 2822 | 2670 | 2570 |
| Correctly Found | 1904 | 1868 | 1833 |
| Missed | 349 | 385 | 420 |
| Misses btn 600_1000 | 91 | 82 | 81 |
| FA | 621 | 505 | 440 |
| MDR | 15,49% | 17,088% | 18,641% |
| FAR | 24,594% | 21,281% | 19,357% |
| PRC | 75,406% | 78,719% | 80,643% |
| RCL | 84,51% | 82,912% | 81,359% |
| F-measure | 79,698% | 80,761% | 80,999% |

| BICFRAME=10sec<br>60/30 ms | Feature Combination [MFCC’s],[Prosodics] III<br>TABLE 5.2.3 | | |
|---|---|---|---|
| THRESHOLD | 440 | 470 | 480 |
| Real Changes | 2253 | 2253 | 2253 |
| Missed | 368 | 398 | 406 |
| FA | 514 | 447 | 420 |
| MDR | 16,333% | 17,665% | 18,020% |
| FAR | 21,425% | 19,417% | 18,526% |
| F-measure | 81,041% | 81,459% | 81,726% |

| BICFRAME=10sec<br>40/20 ms | Feature Combination [MFCC's],[Prosodics] IV<br>TABLE 5.2.4 | |
|---|---|---|
| THRESHOLD | 720 | 710 |
| Real Changes | 2253 | 2253 |
| Missed | 426 | 420 |
| FA | 398 | 415 |
| MDR | 18.908% | 18,641% |
| FAR | 17,887% | 18,460% |
| F-measure | 81,599% | 81,449% |

Feature combination with Prosodics greatly improves system performance, as we can see from table **5.2.1**. On 100_50ms frame there is a **0,2%** improvement in comparison to the same frame of MFCC's implementation(**Table 4.6.1**).

# Chapter 6

## GAUSSIAN MIXTURE MODELS IN BIC

Up to this chapter a single Gaussian was used to fit the data-features yielding a maximum performance of $82{,}2\%$ success on the most efficient characteristics, the baseline MFCC's. Beyond any other feature extraction, feature combination, ideal-frame search, technique utilization, a mixture model is essential to appropriately fit these characteristics and probably catch some missed heterogeneous data or even avoid some insertions (false alarms) due to erroneous model selection. Gaussian mixture densities are a popular representation of non-Gaussian or unknown densities. They constitute a universal function approximation in that, given a sufficient number of components they can approximate any smooth function to arbitrary accuracy. Hence Gaussian mixture model is employed from now on and investigation of how the BIC is going to be applied on new distributions begins.

We already know that BIC criterion needs enough data in order to correctly estimate the event detection of the input signal. This means that, when enough data is given it will detect the change. As a reminder, two seconds at most are required in the implementation to seek the next possible change event from the latter one. This sounds valid in a conversational signal like broadcast news and it facilitates the system, because the initial estimation of the determinant of the covariance matrix and consequently BICvalue needs over 10 frames. Additionally, a minimum number of data are required estimate the number of components of the mixture. The above restriction led to diagonal covariance matrix, which needs at least ten frames (10-dimensional data are extracted) to approximate the real distribution.

Meaning,

$$1\ Component \rightarrow At\ least\ 10\ frames$$

$$2\ Components\ \ \rightarrow At\ least\ 10 * 2\ frames$$

and so on, but in seconds using 50ms frame it means

$$1\ Component \rightarrow 10_{frames} * 50_{ms} = 500_{ms}$$

$$2\ Components \rightarrow 20_{frames} * 50_{ms} = 1_{sec}$$

The above is the minimum possible number of data required to estimate the maximum likelihood parameters of a Gaussian mixture model with k components for data in the n-by-d matrix X, where n is the number of observations and d is the dimension of the data.

Parameters were estimated following the standard Expectation-Maximization (EM) algorithm **[37]**. In some cases, it may converge to a solution which contains singular or close-singular covariance matrix for one or more components. Those components usually contain a few data points almost lying in a lower-dimensional subspace. A solution with singular covariance matrix is usually considered as spurious. Sometimes, this problem may disappear if another set of initial values is applied.

Possible contributing factors include:

a) The number of dimension of data is relatively high, but there are not sufficient observations.
b) Some of the features-variables of data are highly correlated.
c) Some or all the features are discrete.
d) By fitting the data to too many components.

In order to avoid ill-conditioned covariance matrix we used diagonal covariance matrix for each component and added a very small positive number to the diagonal.

Below figures on challenging speaker-change detection are annexed for two to six mixture components distribution.

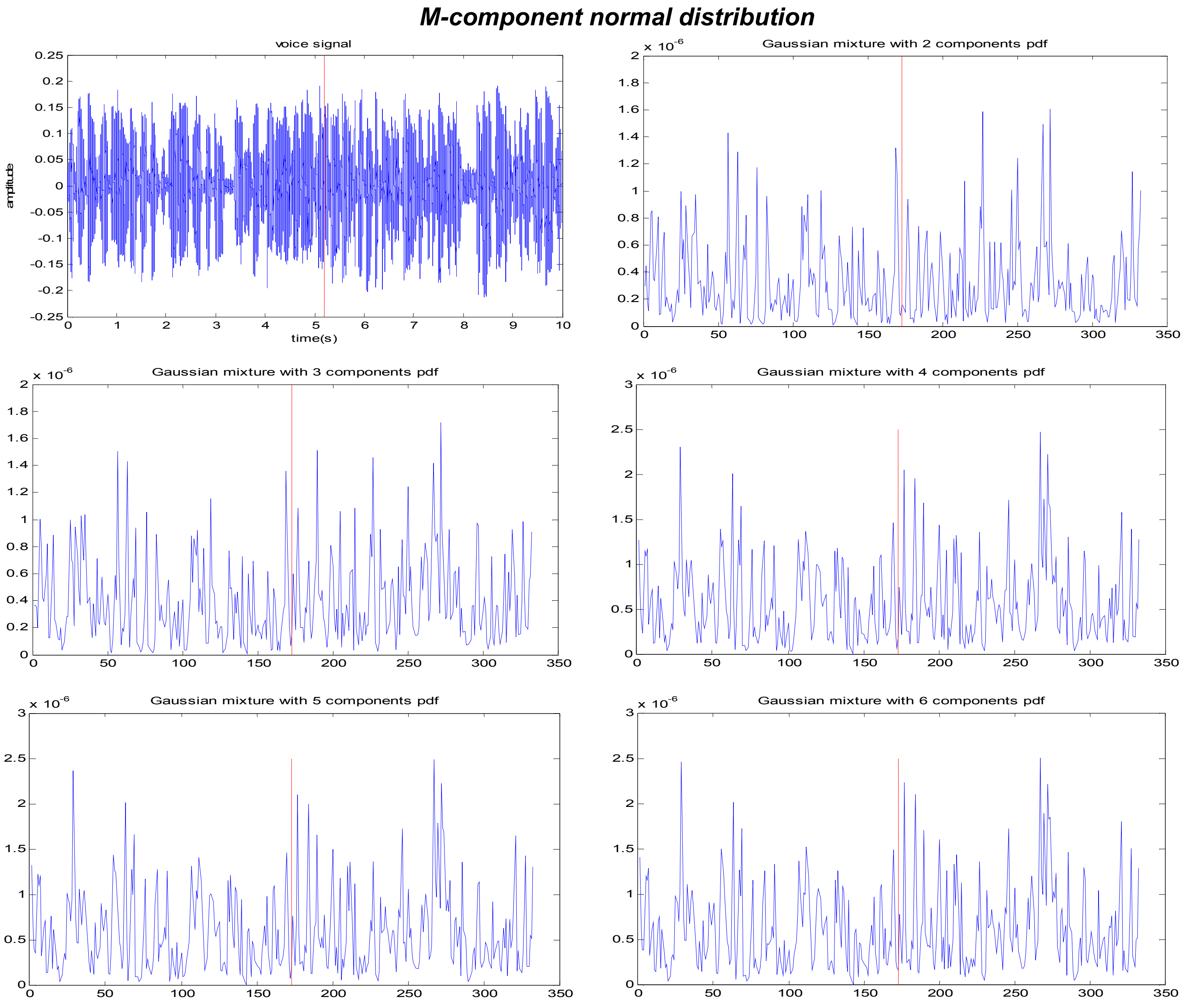


**Figure 4.1** It can be observed that as more components are added, more detail exists. Unfortunately the restriction that was discussed above (using here 30ms frame length) will not allow more than six components.

## 6.1 ENTROPY OF A GAUSSIAN MIXTURE

As shown in chapter **2.3** the maximum likelihood ratio statistic is utilized in Bayesian Information criterion to determine the conformity of the voice signal under consideration.

$$R(i) = N\log|\Sigma| - N_1\log|\Sigma_1| - N_2\log|\Sigma|_2$$
$$\text{minus the model complexity}$$

The logarithm of determinant of the covariance matrix is the entropy for Gaussian density functions and has the above analytic solution. For Gaussian mixtures entropy though, there is no known closed-form solution. Several approximations exist in the international literature, including loose upper and lower bounds, but the only existent approximation that can be demonstrated to converge to the true entropy relies on expensive random sampling methods.

For a continuous random vector **x** ∈ $\mathbb{R}^N$ with probability density function $f(x)$ , the entropy is defined as

$$H(x) = E\{-\log f(x)\} = -\int_{\mathbb{R}^N} f(x)\cdot\log f(x)dx \qquad (6.1)$$

As the entropy is a measure of uncertainty the random vector **x** comprises and it is utilized in many engineering applications. Thanks to their universal approximation property, Gaussian mixtures are a very common representation of density function$(x)$ . $f(x)$ is given by the Gaussian mixture

$$f(x) = \sum_{i=1}^{L} \omega_i \cdot N(x;\mu_i,C_i),$$

where $\omega_i$ are non-negative weighting coefficients with $\sum_i \omega_i = 1$ and $N(x;\mu,C)$ is a Gaussian density with mean vector $\mu$ and covariance matrix $C$.

Due to the logarithm of a sum of exponential function, entropy cannot be estimated in closed form for Gaussian mixtures, with the exception of the case of just a single Gaussian density that we were dealing with until now. In detail entropy is

$$H(x) = \frac{1}{2}\log((2\pi e)^N|C|) \qquad (6.2)$$

On the other hand an approximate solution has to be applied for a mixture as was mentioned above. It is worth mentioning that (6.2) provides an upper bound for all Gaussian mixture random vectors with the same covariance $C$ as in$(6.2)$.

The following notion is used next in order to easily discuss the entropy estimation

$$H(x) = -\int_{\mathbb{R}^N} f(x) \cdot \log g(x) dx \qquad (6.3)$$

$f(x) = g(x)$. This allows differentiating between the Gaussian mixture $g(x)$ that is affected by the logarithm and the Gaussian mixture $f(x)$ that is not argument of the logarithm.

International literature provides numerous methods for an approximate calculation of the entropy for Gaussian mixture random vectors. One of the most straightforward ways to approximate $(6.3)$, results from employing the closed-form solution for a single Gaussian **[38]**. Here, $g(x)$ is replaced by the Gaussian density that exactly captures the first two moments of $f(x)$. Although this method is very efficient, it does not converge to the exact solution. However this method provides an upper bound approximation to the entropy.

The only entropy approximation so far that generally converges to the true entropy and is utilized in our system is given by Monte Carlo sampling. Here, the Gaussian mixture $f(x)$ in $(6.3)$ is represented by a set of samples drawn i.i.d from $f(x)$, which allows a point-wise evaluation of the logarithm term in $(6.3)$. According to the law of large numbers, this approximation converges to the true entropy value as the number of samples goes to infinity. However, a relatively large number of samples have to be used in order to obtain a good approximation, which results in a computationally demanding system. More methods are described in **[39], [40], [41], [42], [43]** and a novel entropy approximation that replaces the logarithm with a multivariate Taylor-series expansion is developed in **[44]**.

Essentially entropy $H(x)$ of probability density function $p(x)$ was approximated as a negative value of the sums of log probabilities of each of the points from the sample of points that are distributed according to the pdf $p(x)$, in an n-dimensional space.

$$H(x) \approx -\frac{1}{N}\sum_{i=1}^{N} \log p(x_i)$$

## 6.2 GAUSSIAN MIXTURE APPLICATION ON BIC AND PERFORMANCE

At this point, we can start the implementation since we have found the method to estimate the entropy of the Gaussian mixture. Experiments begin with two components and all the parameters that may affect the system are under consideration. Every parameter contributes to the alteration of threshold so numerous trials must be done until an ideal setup originates.

Fundamentally, once the MFCC's are extracted, we fit these features on a Gaussian mixture. Then, entropy of this whole segment is estimated and entropy of sub-segments according to the step that we follow to detect events. Yet again the maximum likelihood ratio statistic is used to confirm an event by keeping a maximum estimation of BIC that exceeds an EER posterior-determined threshold.

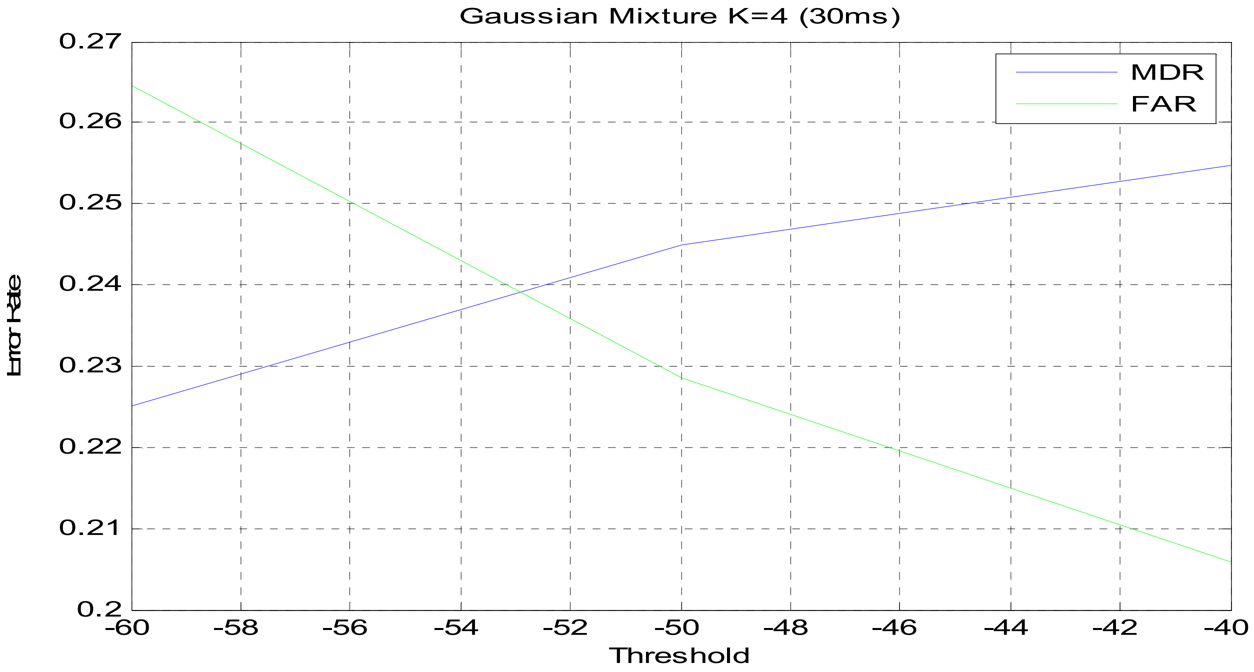
$$R(i) = \frac{1}{N}\sum_{i=1}^{N} \log p(x_i) — \frac{1}{N_1}\sum_{i=1}^{N_1} \log p(x_{1_i}) \; - \frac{1}{N_2}\sum_{i=1}^{N_2} \log p(x_{2_i})$$

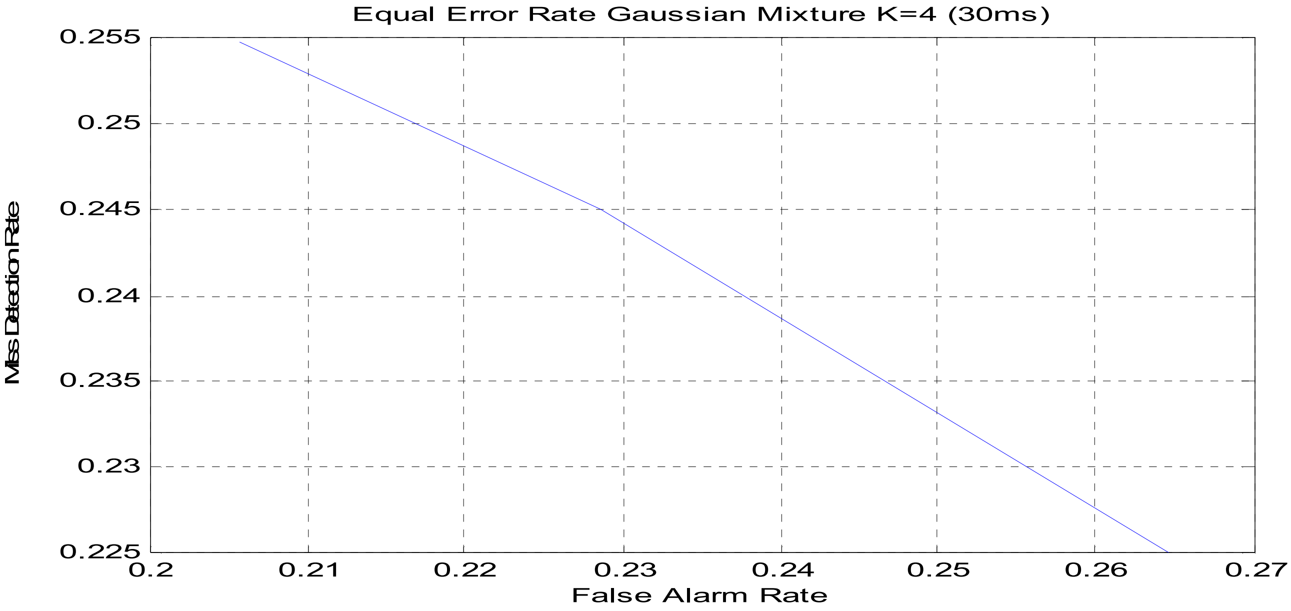
$P = {}^1\!/_2 \cdot K \cdot (d + d) \log N$, K is the number of components

$$BIC(i) = R(i) - P$$

The best results are appended below as the implemented system produced on MFCC's features with 30ms frame.

| BICFRAME=10sec | Gaussian Mixture with four components Results<br>TABLE 6.1 | | |
|---|---|---|---|
| THRESHOLD | -40 | -50 | -60 |
| Real Changes | 2253 | 2253 | 2253 |
| Matlab found | 2407 | 2491 | 2658 |
| Correctly Found | 1679 | 1701 | 1746 |
| Missed | 574 | 552 | 507 |
| Misses btn 600_1000 | 94 | 106 | 99 |
| FA | 435 | 504 | 628 |
| MDR | 25,477% | 24,5% | 22,503% |
| FAR | 20,577% | 22,857% | 26,453% |
| PRC | 79,422% | 77,143% | 73,547% |
| RCL | 74,523% | 75,5% | 77,497% |
| F-measure | 76,894% | 76,312% | 75,47% |




Equal Error Rate Gaussian Mixture K=4 (30ms)
Miss Detection Rate
0.255
0.25
0.245
0.24
0.235
0.23
0.225
0.2
0.21
0.22
0.23
0.24
0.25
0.26
0.27
False Alarm Rate

# Chapter 7

## MAXIMUM LIKELIHOOD LINEAR REGRESSION USE IN SEGMENTATION

Maximum Likelihood Linear Regression (MLLR) estimates linear transformations of automatic speech recognition (ASR) parameters and has achieved significant performance improvements in speaker-independent ASR systems by adapting to target speakers.

In this chapter, we utilize a model adaptation technique which uses a global transformation to tune the Hidden Markov Model (HMM) mean parameters to the new speaker-environment or any of the potential candidate changes in the audio stream. Consequently a new system is built in order to exploit the potentials of maximum likelihood linear regression to further discriminate the inhomogeneous broadcast news audio signals.

The aim of MLLR is to obtain a set of transformation matrices for the model parameters that maximizes the likelihood of the adaptation data. In our case in a limited period of time we need to detect in the whole acoustic signal, whether Gaussian divergence exists, which means that the segment under consideration does not consist of just one Gaussian. Utilizing MLLR we can compare each transformed model to the whole model and measure when the maximum distance occurs (Gaussian divergence, BIC) and record a change detection.

## 7.1. MLLR ESTIMATION FORMULAE

MLLR-based speaker adaptation belongs to the linear transformation family of adaptation algorithms **[45,46,47 and 48]**. Adaptation is performed by linear transformation of the speaker Independent (SI) means and variances of Gaussian distributions of the acoustic model. The approach is reviewed here as presented in **[45,46]**. For example, the adapted Gaussian mean $\hat{\mu}_m$ can be represented as,

$$\hat{\mu}_m = \boldsymbol{W_m}\xi_m \tag{7.1}$$

where $\boldsymbol{W_m}$ is an $n \times (n+1)$ transformation matrix and $\xi_m$ is the extended mean vector,

$$\xi_m = [1\ \mu_m]^T = \left[1\ \mu_{m_1}\mu_{m_2}\right]^T$$

Hence ***W*** can be decomposed into

$$\boldsymbol{W} = [b\ \boldsymbol{A}]$$

where $\boldsymbol{A}$ represents a $n \times n$ transformation matrix and $b$ represents a bias vector.

A typical approach to designing an acoustic model in ASR systems is to use Hidden Markov models (HMMs) to model sub-word units, e.g., tri-phones, with mixture Gaussians distributions modeling the state output distributions. Each individual HMM (for a tri-phone) is usually configured to have 3-states with only left-to-right transitions permitted. The most common solution to training the models are based on maximum likelihood (ML) estimation. A closed form solution for ML estimation of the parameters of the HMMs does not exist. The solution is to use an iterative approach and maximize an auxiliary function, as described by the Baum-Welch algorithm, which is an instance of the Expectation Maximization (EM) algorithm.

The auxiliary function for HMMs can be expressed as

$$\begin{aligned} Q\left(\boldsymbol{M},\widehat{\boldsymbol{M}}\right) &= E_{P(\theta|o)}\left[\log P\left(\boldsymbol{O},\boldsymbol{\Theta}|\widehat{\boldsymbol{M}}\right)|\boldsymbol{O},\boldsymbol{M}\right] \\ &= \textstyle\sum_{\theta\in\boldsymbol{\Theta}} P(\boldsymbol{\Theta}|\boldsymbol{O},\boldsymbol{M}) \log P\left(\boldsymbol{O},\theta|\widehat{\boldsymbol{M}}\right) \end{aligned} \tag{7.2}$$

where, ***M*** is the current model, $\widehat{\boldsymbol{M}}$ is the model being estimated; ***O*** is the entire observation sequence and ***Θ*** represents the set of all possible HMM state sequences ***θ***. It can be shown that finding the $\widehat{\boldsymbol{M}}$, which maximizes the auxiliary function guarantees an increase in the likelihood of the training data ***O***, unless it is already at a maximum

The linear transformation matrix for adaptation of Gaussian mean is estimated from a speaker's acoustic adaptation data using an ML approach and an initial transcription of the adaptation data. Again, the solution is iterative since the state sequence is hidden. The SI Gaussian distributions are grouped into *R* regression classes for the purpose of sharing adaptation transformation $\boldsymbol{W}_r$among them. Considering only the terms that involve the mixture Gaussian distributions, the auxiliary function of Eqn. 7.2, can be written as:

$$Q(\boldsymbol{M},\widehat{\boldsymbol{M}}) = K - \frac{1}{2}\sum_{r=1}^{R}\sum_{c=1}^{C_r}\sum_{m=1}^{M_c}\sum_{\tau=1}^{T}\gamma_m(\tau)\ (o(\tau) - W_r\xi_m)^T\Sigma_m^{-1}(o(\tau) - W_r\xi_m) \quad (7.3)$$

where, K is the normalization constant; $C_r$ is the number of mixture Gaussian distributions in each regression class *r*, and each mixture Gaussian distribution *c* has $M_c$ component Gaussian distributions; $o(\tau)$ is the observation vector at time *τ* and $\gamma_m(\tau)$, $\hat{\mu}_m$ and $\Sigma_m^{-1}$ are the occupation probability at time *τ* , mean vector and inverse covariance of the of the $m_{th}$ Gaussian distribution.

Differentiating Eqn. 7.3, and equating it to 0, the following expression is obtained,

$$\sum_{r=1}^{R}\sum_{c=1}^{C_r}\sum_{m=1}^{M_c}\sum_{\tau=1}^{T}\gamma_m(\tau)\Sigma_m^{-1}o(\tau)\xi_m^T$$

$$= \sum_{r=1}^{R}\sum_{c=1}^{C_r}\sum_{m=1}^{M_c}\sum_{\tau=1}^{T}\gamma_m(\tau)\Sigma_m^{-1}W_r\xi_m\xi_m^T$$

$$= \sum_{r=1}^{R} V^{(r)}W_rD^{(r)} \quad (7.4)$$

where $V^{(r)}$ is the state distribution inverse covariance matrix scaled by the state occupation probability,

$$V^{(r)} = \sum_{t=1}^{T}\gamma_m(\tau)\Sigma_m^{-1} \quad (7.5)$$

and $D^{(r)}$ is the outer product of the extended Gaussian mean vectors,

$$D^{(r)} = \xi_m\xi_m^T \quad (7.6)$$

For the case when the HMM state Gaussian distributions are modeled by a diagonal covariance matrix, a closed form solution for $W_r$ is obtained in the maximization step of the EM algorithm by solving a set of simultaneous equations, one for each row of $W_r$ **[46]**,

$$w_i = G^{(i)-1} z_i^T \qquad (7.7)$$

where $w_i$and $z_i$ are the $i_{th}$ rows of $W_r$and Z respectively. Z is a $n \times (n+1)$ matrix whose elements are given by,

$$z_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)} \qquad (7.8)$$

and the elements of $G^{(i)}$ is given by,

$$g_{jq}^{(i)} = \sum_{r=1}^{R} u_{ii}^{(r)} d_{jq}^{(r)} \qquad (7.9)$$

The EM algorithm guarantees that the adapted Gaussian distribution obtained by applying the transformation matrix $W_r$ will increase the likelihood of adaptation data in each iteration. The row-by-row estimation procedure for $W_r$ can be performed using Gaussian elimination or LU decomposition.

The Gaussian covariance matrices can also be adapted using linear transformations as shown in Eqn. 7.10 or Eqn. 7.11 (proposed in **[49]**),

$$\hat{\Sigma}_m = B_m H_m B_m^T \qquad (7.10)$$
$$\hat{\Sigma}_m = H_m \Sigma_m H_m^T \qquad (7.11)$$

where $B_m$ is the Choleski factor of the original covariance matrix $\Sigma_m$, and $H_m$ is the adaptation transformation matrix in both cases. An iterative estimation procedure for the variance transformation of Eqn. 7.11 that guarantees increase in likelihood of the adaptation data with variance-adapted acoustic model is described in **[49]**. The estimation of variance adaptation is carried out in two steps such that

$$P(O|M) \le P\left(O|\hat{M}\right) \le P\left(O|\tilde{M}\right) \qquad (7.12)$$

where M is the SI model, $\hat{M}$ is the model with the adapted Gaussian mean and $\tilde{M}$ is the model with the adapted Gaussian mean and variance. The adapted covariance matrices are “full”, which can lead to increased computational overhead. A diagonal variance transformation can be estimated by forcing the off-diagonal elements to be zero in the iterative procedure.

Next in this chapter we use the adaptation method that transforms mean vectors.

## 7.2. MLLR IMPLEMENTATION AND APPLICATION ON SEGMENTATION SYSTEM

In order to facilitate the problem of event detection as described in Chapter 2, we will try to adapt this technique on automatic speech segmentation. Our goal is to efficiently utilize the potentials of MLLR, as it is able to adapt signal's features in any environment and to any speaker observed, altering the parameters of their distribution. In essence, the initial idea is to estimate a transformation matrix call it *H* for the frame in which we search a speaker-change. This matrix will update the mean of the whole signal using7.1, as discussed in the previous entity, which means that we now have on this particular frame its Gaussian's parameters, at a maximum likelihood, re-estimated. Then in turn we choose a decreasing, at one hundred milliseconds, segment in the frame we search the speaker change and compute a corresponding transformation matrix call it *H_L*.

By means of the powerful Hidden Markov Toolkit and Matlab environment an interface was built and we implemented this new system. Feature vectors are extracted in HTK (Mfcc's at 20/10 milliseconds rate) from a broadcast news signal. Then we train an HMM with a single mixture and a global mean transformation, without regression classes as there is just one Gaussian, is estimated. Subsequently, we enter arguments into Matlab recursively for every frame and segments (defined in Matlab scripts) tested in each frame.

A second metric is introduced to exploit MLLR transformation. This metric will use *H* to transform the mean of the whole signal's mean resulting at maximum likelihood estimation parameters of the segment we investigate. It will also use *H_L* progressively, by the same reasoning for the decreasing subsegment, but this matrix is estimated between the segment's features and the corresponding's frame features.

Before proceeding, we need to introduce a reminder for multivariate Gaussian densities. Let

$$\{N_j(x;\mu_j,\Sigma_j)\in G_d|x,\mu\in\mathbb{R}^d,\Sigma_j\in S_{++}^{d\times d},j=1{,}2\}$$

be two Gaussian densities where

$$N_j(x;\mu_j,\Sigma_j)=|2\pi\Sigma_j|^{-\frac{1}{2}}exp\left\{-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right\},$$

$|\cdot|$ is the determinant, $\mu_j$ is the mean vector, $\Sigma_j$ is the covariance matrix and $S_{++}^{d\times d}$ is the space of real symmetric positive semi definite matrices (or nonnegative-definite).

The symmetric KL divergence is based on Kullback's measure of discriminatory information:

$$I(P_1,P_2)=-\int_{\varepsilon}p_1\log(p_1/p_2)dx.$$

Kullback realizes the asymmetry of $I(P_1,P_2)$ and describes it as the directed divergence. To achieve symmetry, Kullback defines the divergence as:

$$I(P_1,P_2)+I(P_2,P_1)$$

and notes that it is positive and symmetric but violates the triangle inequality **[50].** Hence, it cannot define a metric structure. The closed form expression for the symmetric KL divergence between $\mathcal{N}_1$ and $\mathcal{N}_2$ can be written as:

$$d_{KL}(\mathcal{N}_1,\mathcal{N}_2)=\frac{1}{2}(\mu_1-\mu_2)^T(\Sigma_1^{-1}+\Sigma_2^{-1})(\mu_1-\mu_2)+\frac{1}{2}tr(\Sigma_1^{-1}\Sigma_2+\Sigma_2^{-1}\Sigma_1-2I)\quad(5.14)$$

where $tr$ is the matrix trace (sum of the diagonal elements-eigenvalues of the matrix) and $I$ is the identity matrix. Equation $(5.14)$ describes $d_{KL}$ as a sum of two components, one due to the difference in means weighted by the covariance matrices, and the other due to the difference in variances and covariance matrices **[50]**. If $\Sigma_1=\Sigma_2=\Sigma$, then $d_{KL}$ expresses the difference in means which is the exact form of the Mahalanobis distance:

$$(\mu_1-\mu_2)^T(\Sigma^{-1})(\mu_1-\mu_2)\qquad(5.15)$$

However, if $\mu_1=\mu_2=\mu$, then $d_{KL}$ expresses the difference or the dissimilarity between covariance matrices $\Sigma_1$ and $\Sigma_2$:

$$d_{KL}(\mathcal{N}_1,\mathcal{N}_2)=\frac{1}{2}tr(\Sigma_1^{-1}\Sigma_2+\Sigma_2^{-1}\Sigma_1-2I).$$

According to the above we use Gaussian Divergence through $(5.15)$ and estimate as follows:

$$GD(i) = \frac{1}{2}(H\mu - H_L\mu)^T([\Sigma]^{-1})(H\mu - H_L\mu)$$

The above expression estimates every 100 milliseconds the distance between two Gaussians and the maximum value over a threshold is recorded as event detection.

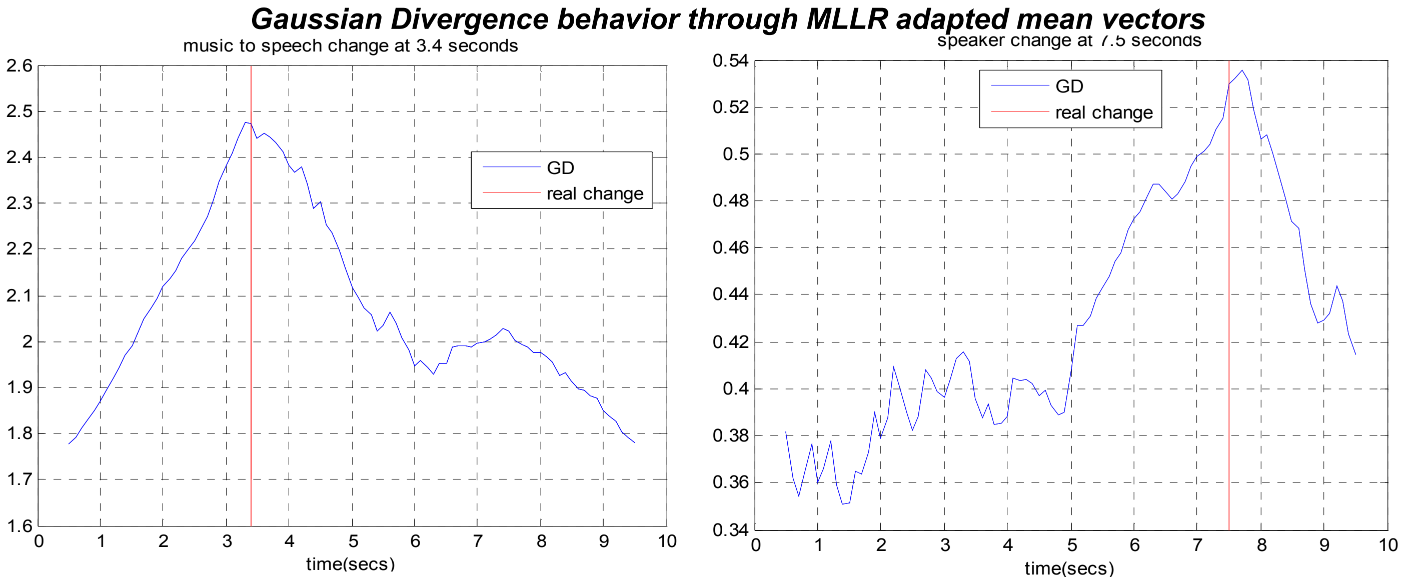


**Figure 7.1** The first figure shows correct event detection. Also the second figure shows a correct speaker change. These two figures show the difficulty to reach to an "ideal" threshold.

Observing the potentials of Gaussian Divergence, a combination of metrics tested until now is pending. $Mahalanobis\ Distance$ cannot ensure a threshold to yield the best possible results and thus we utilize this metric to emphasize event detection from our known Bayesian Information Criterion. All systems in this chapter are based on 20millisecond with 10 milliseconds overlap frame and ten dimensional features. This extraction feature window on BIC produces numerous false alarms. Consequently, to tackle these drawbacks our proposal occurs as:

$$Metric_{\ combination} = BIC(i) * GD(i)$$

Above equation is expected to ensure the existence of a threshold of this maximum likelihood ratio statistic, which will ensue to better rate of false alarms and missed detections.

| BICFRAME=10sec<br>20/10 ms | MLLR GD*BIC<br>TABLE 7.1 |
|---|---|
| THRESHOLD | 32000 |
| Real Changes | 2253 |
| Missed | 366 |
| FA | 623 |
| MDR | 16,245% |
| FAR | 24,82% |
| F-measure | 79,236% |

The above results are very promising as MDR fell to 16,245% while False Alarm rate remained at 24,82% at this specific frame length extraction. On previous modules it is expected to produce  over 40% FAR. MLLR may efficiently contribute to the video segmentation problem.

# CHAPTER 8

## Conclusions and future work

As explained in the introduction, characterizing numerous speakers in a broadcast signal has always been difficult. Current knowledge, existing procedures and experimental applications were examined in this thesis.

Specifically, an attempt was made to facilitate the identification, classification, clustering and by extrapolation the extraction of particular segments of speech. Furthermore, it is now possible to recreate segments which belong to different categories (e.g. music, speech, prerecorded advertisements, etc.).

In the effort to overcome the obstacles encountered, international literature provided current algorithms and thus a temporary system was designed and initial experiments were carried out, allowing for ongoing adjustments and the ironing out of technical hitches, before finally reaching the clearest and most effective method.

In order to represent sound more clearly, several audio characteristics that warp the frequency were experimented with. The number of coefficients (dimensions) was calculated taking into consideration the drawbacks of the algorithm used and subsequently the ideal frame-window according to which the coefficients were extracted was meticulously deliberated upon.

Sound coefficients and their yields were thoroughly tested and through trial and error, a variation on the BIC procedure was devised particularly to enable the parallel analysis of two different features simultaneously.

As the BIC method defines, it is assumed that the acoustic feature vectors in each of the two audio segments are drawn from a Gaussian distribution and a change detection results from the dissimilarity of the Gaussians. Indeed, we were able eventually, to deduce whether a dataset is modeled by two Gaussians or just one Gaussian. We concluded that by implementing a Gaussian mixture, each signal would be more correctly modeled. Thus, instead of calculating the entropy of each section's Gaussian, we calculate the approximate entropy of the Gaussian mixture of every section.

We introduced the idea of maximum likelihood linear regression into the problem of automatic video segmentation, which is capable of shifting the assumed Gaussian distribution to the "correct" mean and to convert to the "correct" covariance matrix, starting from an initial estimation.

Our variation on the BIC is a novel approach. We encountered many obstacles and overcame numerous problems. The crux of the matter is that through implementing the variation, we are now able to identify, analyze pattern distribution and recreate the sound signal. The final formula, at which we arrived, shifts the mean and transforms the covariance matrix of a normal distribution and used as a metric the Gaussian Divergence, in conjunction with the "original" BIC.

Quite simply, the study of the distribution patterns of sound files led us to the creation of a functional segmentation and clustering system, for application in real terms from news broadcasts. Aside from the aforementioned application, a system such as this could widely be applied in the creation of an entire data base, which would facilitate the automatic search for the speech pattern of a particular speaker, presenter, journalist, artist or politician etc. Particularly in conjunction with the ASR system, even a topic search could be enabled. We showed that the implementation of the various methods works in practice. Our best results were achieved when the error rate was brought down to 17%. This system can deliver results in almost any application that requires speaker segmentation. The algorithms and the new ideas which were described could form the basis for a new, more detailed and broader system of segmentation and clustering that will essentially accelerate and improve the results of programs that focus on speech and the analysis and reproduction of audio signals.

# REFERENCES

[1] Pedro J. Moreno, Jean Manuel Van Thong, Beth Logan, Blair Fidler, Katrina Maffey, Matthew Moore. SPEECHBOT: A Content-Based Search Index for Multimedia on the WEB.

[2] D.Dimitriadis, P. Maragos, and A.Potamianos. Auditory teager energy cepstrum coefficients for robust speech recognition. Proc. European Conf. on Speech Communication and Technology — Interspeech 2005, Lisbon, Portugal, pages pp.3013-3016, 2005.

[3] P. Maragos, J. F. Kaiser, and T. F. Quatieri. Energy separation in signal modulations with application to speech signals. IEEE Transactions on Signal Processing, Vol.41:pp.3024-3051, 1993.

[4] H. Gish and M. Schmidt, “Text-independent speaker identification,” IEEE Signal Processing

[5] M. Siegler, U. Jain, B. Raj, and R. Stern, “Automatic segmentation, classification and clustering of broadcast news audio,” in *Proc. DARPA Speech Recognition Workshop*, Feb. 1997, pp. 97–99.*Mag.*, vol. 11, no. 4, pp. 18–32, 1994.

[6] M. Basseville, “Distance measures for signal processing and pattern recognition,” *Eur. J. Signal Process.*, vol. 18, no. 4, pp. 349–369, Dec. 1989.

[7] L. Lu and H.-J. Zhang. Real-time unsupervised speaker change detection. ICPR, 2002.

[8] K. Joergensen, L. Moelgaard, and L. K. Hansen. Unsupervised speaker change detection for broadcast news segmentation. Technical Report, nformatics and Mathematical Modelling, Technical University of Denmark.

[9] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel. Strategies for automatic segmentation of audio data. Proc. IEEE Int.Conf Acoust.,Speech, Signal Process., 2000.

[10] J. Ajmera, I. McCowan, and H. Bourlard. Robust speaker change detection. IEEE Signal Processing Letters, Vol.11:pp.649{651, 2004.

[11] S. Kwon and S. Narayanan. Speaker change detection using a new weighted distance measure. ICSLP, 2002.

[12] R. Bakis, S. Schen, P. Gopalakrishnan, R. Gopinath, S. Maes, and L. Polymenakos, “Transcription of broadcast news shows with the IBM large vocabulary speech recognition system,” in *Proc. IEEE ICASSP-97: Int. Conf. Acoust., Speech, and Signal Proc.*, Munich, Germany, 1997,pp. 711–714.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, Chapter 3. John Wiley & Sons, Inc., 2 edition, 1997.

[14] T. Hain, S. E. Johnson, A. Tuerk, P. C.Woodland, and S. J. Young, “Segment generation and clustering in the HTK broadcast news transcription system,” in *Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, 1998, pp. 133–137.

[15] J.-L. Gauvain, L. Lamel, and G. Adda. The limsi broadcast news transcription system. Speech Communication, Vol.37:p.89-108, 2002.

[16] S. Chen and P. Gopalakrishnan, “Speaker, environment and channel change detection and clustering via the Bayesian information criterion,” in *Proc. DARPA Broadcast News Transcription Understanding Workshop*, Feb. 1998, pp. 127–132.

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

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

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

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

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

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

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

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

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

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

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

[28] S. Cheng and H. Wang, “Metric SEQDAC: A hybrid approach for audio segmentation”, in Proc. *8th Int. Conf. Spoken Language Processing*, Jeju, Korea, October 2004.

[29] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain. Multistage speaker diarization of broadcast news. IEEE Trans. Audio, Speech and Lang. Process., Vol.14:pp.1505-1512, 2006.

[30] J. Ajmera, I. A. McCowan, and H. Bourlard. Robust hmm-based speech/music segmentation. Proc. IEEE Int.Conf Acoust.,Speech, Signal Process., Vol.1:pp.297-300, 2002.

[31] L. Lu, H.-J. Zhang, and H. Jiang. Content analysis for audio classification and segmentation. IEEE Trans., Speech, Audio Process., Vol.10:pp.504-516, 2002.

[32] Discrete-Time Speech Signal processing\Quatieri

[33] *RASTA_PLP* David Burton Last updated: 9/22/98 An ESPS program for robust speech-recognition feature computation

[34] Approaches to Language Identification using Gaussian Mixture Models and Shifted Delta Cepstral Features*Pedro A. Torres- arrasquillo1, 2, Elliot Singer2, Mary A. Kohler3,Richard J. Greene2, Douglas A. Reynolds2, and J.R. Deller, Jr.1*

[35] MODELING PROSODIC DYNAMICS FOR SPEAKER RECOGNITION Andre G. Adami, Radu Mihaescu, Douglas A. Reynolds, John J. Godfrey

[36] A comparative Performance Study of Several Pitch Detection Algorithms\Rabiner, Cheng, IEEE tran. on acoustics, speech and sig. Proc., 1976

[37] EXPECTATION MAXIMIZATION: A GENTLE INTRODUCTION ,Moritz Blume

[38] J. R. Hershey and P. A. Olsen, “Approximating the Kullback-Leibler Divergence between Gaussian Mixture Models,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 4, Apr. 2007, pp. IV–317–IV–320.

[39] D. E. Catlin, Estimation, Control, and the Discrete Kalman Filter, 1st ed., ser. Applied Mathematical Sciences. New York: Springer-Verlag, 1989, vol. 71.

[40] J. Goldberger, S. Gordon, and H. Greenspan, “An Efficient Image Similarity Measure based on Approximations of KL-Divergence Between Two Gaussian Mixtures,” in Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 1, Oct. 2003, pp. 487–493.

[41] S. J. Julier and J. K. Uhlmann, “A New Extension of the Kalman Filter to Nonlinear Systems,” in International symposium on Aerospace/Defence Sensing, Simulation and Control, 1997.

[42] J. W. Fisher III and J. C. Principe, “A methodology for information theoretic feature extraction,” in Proceedings of the IEEE International Joint Conference on Neural Networks, A. Stuberud, Ed., vol. 3, 1998, pp. 1712–1716.

[43] J. W. Fisher and T. Darrell, “Speaker Association With Signal-Level Audiovisual Fusion,” IEEE Transactions on Multimedia, vol. 6, no. 3, pp. 406–413, Jun. 2004.

[44] On Entropy Approximation for Gaussian Mixture Random Vectors Marco F. Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D. Hanebeck

[45] V. Digalakis, D. Rtischev, and L. Neumeyer, “Speaker adaptation using constrained estimation of Gaussian mixtures," IEEE Transactions on Speech and Audio Processing,vol. 3, no. 5, pp. 357-366, 1995.

[46] Leggetter, C.J., Woodland, P.C. 1995. Maximum likelihood linear regression for speaker adaptation of HMMs. Computer Speech & Language 9, 171–185.

[47] J. Neto, L. Almeida, M. M. Hochberg, C. Martins, L. Nunes, S. J. Renals, and A. J. Robinson, “Unsupervised speaker-adaptation for hybrid HMM-MLP continuous speech recognition systems," in Proc. of Eurospeech, vol. 1, 1995, pp. 187-190.

[48] L. Neumeyer and M. Weintraub, “Probabilistic optimum filtering for robust speech recognition," in Proc. of ICASSP, vol. 1, 1994, pp. 417-420.

[49] M. Gales and P. Woodland, “Mean and variance compensation within the MLLR framework," Computer Speech & Language, vol. 10, pp. 249-264, 1996.

[50] Kullback, S.: Information Theory and Statistics – Dover Edition. Dover, New York (1997)